

Treball de Fi de Grau

Enginyeria en Tecnologies Industrials

Mecanisme per aïllar una instal·lació durant una tempesta elèctrica

MEMÒRIA

Autor: Eloi Codina Torras
Director: Lluís Solano Albajes
Convocatòria: Gener 2020



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

En aquest projecte es proposa un prototip per aïllar dues instal·lacions elèctriques a fi d'evitar la transmissió de sobrevoltatges derivats de descàrregues atmosfèriques. Es tracta d'una aproximació a l'enginyeria des d'una visió pràctica a partir d'un problema trobat durant el disseny de l'alimentació d'un repetidor Wi-Fi ubicat a dalt d'una muntanya.

Per començar, s'ha calculat a quina distància s'han de trobar els terminals d'unió entre les dues instal·lacions durant una tempesta elèctrica per tal que l'aire, en condicions normals, sigui un medi suficientment aïllant per evitar la creació d'arcs elèctrics que transmetrien el sobrevoltatge, malmetent electrodomèstics i instal·lacions.

A continuació, s'ha dissenyat i construït el dispositiu físic, seleccionant el material idoni i els sensors necessaris pel seu funcionament i buscant la màxima eficiència energètica del controlador del dispositiu, programat en Python en una Raspberry Pi.

S'ha posat especial èmfasi en la gestió de projectes informàtics amb la filosofia *Agile*, que permet adaptar el producte final a mesura que avança el projecte i ajuda a tenir una versió funcional en poc temps.

El resultat final és un mecanisme autònom però que disposa d'una interfície en forma de pàgina web per al seu control manual en cas de necessitat.

Sumari

SUMARI	4
ÍNDIX DE FIGURES	7
ÍNDIX DE TAULES	10
1. GLOSSARI	11
2. PREFACI	15
3. INTRODUCCIÓ	17
3.1. Objectius del projecte	17
3.2. Abast del projecte	17
4. ANÀLISI DE LA PROBLEMÀTICA	18
4.1. Descàrregues elèctriques atmosfèriques.....	18
4.2. Efectes d'un llamp sobre la instal·lació.....	20
4.3. Dispositius de protecció existents.....	21
5. VISIÓ GLOBAL DEL PROTOTIP	23
6. DISSENY I CONSTRUCCIÓ DEL PROTOTIP	26
6.1. Elecció del material.....	26
6.2. Estructura i sistema mòbil.....	27
6.3. Terminals elèctrics	29
6.4. Ubicació dels components.....	30
7. COMPONENTS ELECTRÒNICS	32
7.1. Raspberry Pi	32
7.1.1. Paràmetres elèctrics de la Raspberry Pi	32
7.1.2. Característiques físiques.....	35
7.1.3. Sistema operatiu i llenguatge de programació	36
7.1.4. Alternatives a la Raspberry Pi.....	37
7.2. Sensor ultrasònic de distància SRF005.....	38
7.2.1. Funcionament i càlcul de la distància.....	38
7.2.2. Filtratge de les dades obtingudes	40
7.2.3. Connexió a la Raspberry Pi i control amb Python	43
7.3. Controlador del motor: L293D.....	44
7.3.1. Modulació per amplada de polsos	45
7.3.2. Connexió a la Raspberry Pi i control amb Python	46

7.4.	Sensor de llamps: AS3935.....	46
7.4.1.	Funcionament	47
7.4.2.	Propietats del sensor	48
7.4.3.	Connexió amb la Raspberry Pi	49
7.4.4.	Programació amb Python.....	50
8.	SISTEMA INFORMÀTIC	51
8.1.	Components principals	51
8.1.1.	Control del GPIO: <i>pigpio</i>	51
8.1.2.	Django	52
8.1.2.1.	Django REST Framework	53
8.1.2.2.	Aplicacions	55
8.1.3.	Base de dades: PostgreSQL.....	56
8.1.4.	Gestió de tasques asíncrones: <i>Celery</i> i <i>Redis</i>	57
8.2.	Aplicacions	59
8.2.1.	API	59
8.2.1.1.	Models.....	59
8.2.1.2.	Rutes.....	60
8.2.2.	Pàgina web	63
8.2.2.1.	Targetes disponibles	64
8.2.2.2.	Creació d'una plantilla <i>HTML</i>	67
8.2.2.3.	Control de la pàgina	68
8.3.	Tasques en segon pla	68
8.3.1.	motor_background	68
8.3.2.	send_email.....	68
8.3.3.	force_status_update.....	68
8.3.4.	download_weather_forecast	69
8.3.5.	check_online_lightning	69
8.3.6.	as3935_monitor	69
8.3.7.	reconnect_motor_timer	69
8.4.	Funcionament del sistema	70
9.	GESTIÓ DEL PROJECTE	71
9.1.	Programes emprats durant el desenvolupament	71
9.1.1.	PyCharm.....	71
9.1.2.	Connexió SSH: MobaXterm.....	71
9.1.3.	Control de versió: Git	71
9.2.	Gestió global del projecte	72
9.3.	Gestió de la part informàtica: <i>Agile</i> i <i>Scrum</i>	73
9.3.1.	Eines utilitzades	73

9.3.1.1. YouTrack.....	73
9.3.1.2. KanBan	74
9.3.2. Estructura de les tasques.....	74
9.3.3. Flux de treball	76
9.3.4. Iteracions i litúrgies	77
10. ANÀLISI I MITIGACIÓ DE RISCS	79
11. PRESSUPOST	81
11.1. Cost del material	81
11.2. Cost del personal	82
11.3. Cost de l'equipament	83
11.4. Cost energètic.....	83
11.5. Cost total.....	84
12. IMPACTE AMBIENTAL	85
12.1. Consum energètic.....	85
12.2. Materials	85
12.3. Components electrònics	86
CONCLUSIONS	87
AGRAÏMENTS	89
BIBLIOGRAFIA	91

Índex de figures

Figura 4.1: llamp núvol-terra. Font: Ron Rev Fenomeno, pixabay.com.....	18
Figura 4.2: distribució de les càrregues en un núvol. Font: Enciclopedia Britannica.....	18
Figura 4.3: procés d'un llamp núvol-terra. Font: Martin Uman (1987).....	19
Figura 4.4: esquema de la instal·lació d'un SPD genèric. Font: Standler (2002).....	22
Figura 5.1: components principals. Font: pròpia.....	23
Figura 5.2: procés de desconexió del sistema. Font: pròpia.....	23
Figura 5.3: procés de connexió. Font: pròpia	24
Figura 5.4: inici del moviment del mecanisme. Font: pròpia	24
Figura 5.5: final del moviment. Font: pròpia.....	25
Figura 5.6: peticions a través de l'API. Font: pròpia	25
Figura 6.1: aproximació del sistema motor-corretja-tensor. Font: pròpia.....	27
Figura 6.2: extrem del dispositiu amb forma de pont. Al seu interior s'ubicarà el motor i el sensor ultrasònic de distància. Font: pròpia	27
Figura 6.3: extrem del dispositiu amb forma de pont. Al seu interior s'ubicarà el tensor de la corretja i la part vertical contindrà els terminals elèctrics del cable de l'antena. Font: pròpia	27
Figura 6.4: part mòbil. Font: pròpia.....	28
Figura 6.5: exemple de connectors plans. Font: amazon.es	29
Figura 6.6: terminal elèctric amb molla.. Font: pròpia	29
Figura 6.7: terminal elèctric pla. Font: pròpia.....	29
Figura 6.8: exemple de terminal elèctric. Font: amazon.es	30
Figura 6.9: prototip dissenyat on es poden veure tots els components. Font: pròpia	30
Figura 6.10: prototip construït, on es poden identificar tots els components descrits en aquest capítol. Font: pròpia.....	31

Figura 7.1: estimació de la producció d'energia. Font: Wattia	33
Figura 7.2: esquema de la Raspberry Pi 3B. Font: raspberry.org.....	35
Figura 7.3: disposició dels pins GPIO. Font: raspberry.org	36
Figura 7.4: SRF005. Font: Pròpia.....	38
Figura 7.5: comparació gràfica de la distància calculada pel sensor SRF005 amb o sense filtre. Font: pròpia	40
Figura 7.6: zona de detecció del SRF005. Font: hobbizine.com	41
Figura 7.7: modelització del problema per conèixer la posició real de la part mòbil. Font: pròpia	41
Figura 7.8: divisor de tensió per aconseguir 3,3 V a partir d'una font de 5 V. Font: pròpia ...	43
Figura 7.9: captura de pantalla de piscope amb dades de l'SRF005. Font: pròpia	44
Figura 7.10: L293D. Font: pròpia.....	44
Figura 7.11: diagrama de blocs del L293D. Font: ti.com	45
Figura 7.12: ona de polsos d'amplada variable. Font: rudysarduinoprojects.wordpress.com	45
Figura 7.13: MOD-1016. Font: pròpia.....	47
Figura 7.14: diagrama d'aplicació de l'AS3935. Font: ams.com	47
Figura 7.15: eficiència de detecció per diferents valors de WIDTH amb SREJ=0000. Font: ams.com	48
Figura 7.16: protocol I ² C. Font: i2c.info	49
Figura 8.1: exemple de model. Font: pròpia	52
Figura 8.2: procés d'una petició en Django. Font: pròpia	53
Figura 8.3: exemple de JSON. Font: pròpia	54
Figura 8.4: exemple de serialitzador. Font: pròpia.....	55
Figura 8.5: comunicació entre PostgreSQL i Django. Font: pròpia.....	57

Figura 8.6: funcionament de la cua al Celery. Font: pròpia	58
Figura 8.7: capçalera de la pàgina web. Font: pròpia	63
Figura 8.8: control del mecanisme des de la pàgina web. Font: pròpia	63
Figura 8.9: dades històriques mostrades a la pàgina web. Font: pròpia	64
Figura 8.10: peu de la pàgina web. Font: pròpia	64
Figura 8.11: targeta "Últim llamp" de la pàgina web. Font: pròpia	64
Figura 8.12: targeta "predicció avui" de la pàgina web. Font: pròpia	65
Figura 8.13: targeta "Propera tempesta" de la pàgina web. Font: pròpia	65
Figura 8.14: opcions de la targeta "Reconnexió sistema" de la pàgina web. Font: pròpia....	65
Figura 8.15: targeta "Estat actual" de la pàgina web. Font: pròpia	66
Figura 8.16: targeta "Control manual" quan el sistema es troba connectat. Font: pròpia	66
Figura 8.17: targeta "Control manual" quan el sistema es troba desconnectat. Font: pròpia	67
Figura 8.18: exemple de requadre mostrat per la pàgina web amb informació sobre la petició. Font: pròpia	68
Figura 9.1: diagrama de Gantt. Font: pròpia.....	72
Figura 9.2: captura de pantalla de YouTrack. Font: pròpia.....	74
Figura 9.3: KanBan utilitzat durant el desenvolupament. Font: pròpia.....	74
Figura 9.4: estructura de les tasques. Font: pròpia	75
Figura 9.5: exemple de User Story. Font: pròpia	76
Figura 9.6: flux de treball en Agile. Font: pròpia	76
Figura 9.7: una iteració. Font: pròpia	78
Figura 12.1: producció d'energia elèctrica a l'estat espanyol. font: ree.es	85

Índex de taules

Taula 6.1: propietats de la instal·lació elèctrica de la masia. Font: Wattia.....	33
Taula 6.2: consum dels sensors connectats. Font: pròpia.....	34
Taula 6.3: comparació entre Arduino Uno i Raspberry Pi. Font: pròpia.....	37
Taula 6.4: variables que intervenen en l'equació de la velocitat del so segons la temperatura del medi. Font: University Physics.....	39
Taula 6.5: velocitats del so depenent de la temperatura de l'habitació. Font: pròpia.....	39
Taula 6.6: distància calculada amb velocitat del so a temperatura de referència. Font: pròpia	40
Taula 7.1: comparació entre PostgreSQL i SQLite. Font: TablePlus.....	56
Taula 8.1: hores dedicades a cada activitat del projecte. Font: pròpia	72
Taula 10.1: cost del material. Font: pròpia	82
Taula 10.2: cost del personal. Font: pròpia	83
Taula 10.3: cost de l'equipament. Font: pròpia.....	83
Taula 10.4: cost energètic. Font: pròpia	84
Taula 10.5: cost total. Font: pròpia	84

1. Glossari

- **Agile:** filosofia de gestió de projectes, especialment informàtics, en el qual es busca adaptar-se als canvis de forma fàcil i intenta aconseguir la màxima participació del client fent entregues de funci'onalitats de forma regular.
- **API:** protocol de comunicació entre programes.
- **API REST:** protocol de comunicació entre programes seguint l'arquitectura REST, tal i com està definida a l'apartat 8.1.2.1.
- **Caché:** memòria temporal.
- **Dimoni:** programa que s'executa en segon pla de forma contínua.
- **Django:** entorn de treball web.
- **DRF:** acrònim de *Django REST Framework*. Mòdul de *Django* que permet crear *API RESTs*.
- **Entorn de treball:** plataforma que permet desenvolupar programari fàcilment, ja que incorpora funcionalitat genèrica que es pot personalitzar segons l'aplicació
- **Epic:** nivell més general en la definició de tasques seguint *Agile*.
- **Execució asíncrona:** codi que s'executa en segon pla, permetent l'execució d'altres funcions que no depenen de la finalització de la tasca asíncrona.
- **Feature:** segon nivell en la definició de tasques seguint *Agile*.
- **HDMI:** *High-Definition Multimedia Interface*. Interfície per transmetre àudio i vídeo.
- **HTML:** *HyperText Markup Language*. Llenguatge que permet estructurar documents per tal de ser mostrats en pàgines webs a través de navegadors.
- **HTTP:** *HyperText Transfer Protocol*. Protocol que defineix com s'han de transmetre missatges a través de la web.
- **Hypermedia:** extensió de l'HTML que permet incloure àudio, imatges, vídeos o enllaços web interactius.
- **Impedància:** mesura de l'oposició a la circulació de corrent en un circuit elèctric.

- **JSON:** *JavaScript Object Notation*. Especificació d'un format de dades per transmetre-les entre dispositius diferents.
- **Llenguatge interpretat:** llenguatge de programació on el codi es tradueix en instruccions que pot entendre la màquina a mesura que es va executant. És el contrari de compilat, on primer es tradueix el codi desenvolupat en llenguatge de màquina i a continuació s'executa.
- **Microcontrolador:** circuit integrat programable.
- **Mòdul (Python):** fitxer que conté codi de *Python* (defineix funcions, classes, etc) i que es pot importar en diferents *scripts* per tal de reutilitzar codi.
- **MOV:** varistor d'òxid metàl·lic. Tipus d'SPD.
- **Permís de superusuari:** tipus de permís màxim que pot aconseguir un usuari en una màquina. Permet eliminar qualsevol fitxer o modificar qualsevol paràmetre del sistema.
- **Petició DELETE:** mètode d'HTTP pel qual s'informa al servidor que es vol eliminar el recurs indicat en la ruta.
- **Petició GET:** mètode d'HTTP pel qual s'informa al servidor que es vol una còpia de la informació que hi ha a la ruta indicada.
- **Petició POST:** mètode d'HTTP pel qual s'informa al servidor que es vol crear un nou recurs a la ruta indicada.
- **Petició PUT:** mètode d'HTTP pel qual s'informa al servidor que es vol modificar el recurs indicat en la ruta.
- **Pin GPIO:** pin que permet transmetre dades (*General Purpose Input-Output*). En el cas de la Raspberry Pi únicament es permeten dades digitals (nivell alt o nivell baix).
- **Script:** fitxer que conté codi executable.
- **Scrum:** metodologia que segueix la filosofia *Agile*.
- **SMC:** inicials de Servei Meteorològic de Catalunya.
- **SO:** inicials de Sistema Operatiu.

- **SPD:** protector de sobretensions.
- **SSH:** protocol que permet administrar sistemes a través de xarxes.
- **Stateless (arquitectura REST):** el servidor no necessita informació obtinguda en peticions anteriors per tal de respondre el que ha demanat el client.
- **Tensió de ruptura:** voltatge mínim a partir del qual un material aïllant es converteix en conductor.
- **Transductor:** dispositiu que converteix una magnitud física en una altra. Per exemple, energia elèctrica en so.
- **URL:** *Uniform Resource Locator*. Adreça que especifica la ubicació d'un recurs a la web.
- **User Story:** nivell més específic en la definició de tasques seguint *Agile*.
- **Vista (Django):** funció de *Python* que, a partir d'una petició d'un client, executa una sèrie d'operacions lògiques i retorna cert contingut (per exemple *HTML*).

2. Prefaci

Quan vaig decidir estudiar Enginyeria en Tecnologies Industrials tenia clar que el Treball Fi de Grau havia de ser eminentment pràctic i útil. És per aquest motiu que quan vaig trobar un tema que complia aquests requisits no vaig dubtar d'endinsar-m'hi.

Durant l'estiu de l'any 2018, juntament amb la família, vam aconseguir connectar una masia aïllada d'Osona a Internet. Un projecte de gran envergadura, ja que a l'habitatge no arriba connexió de telèfon ni hi ha visibilitat a cap repetidor, però dut a terme de forma satisfactòria: s'ha aconseguit tenir una velocitat de fins a 100 Mbps, similar a algunes ofertes de fibra òptica d'algunes companyies.

Aquesta connexió és per mitjà d'un radioenllaç Wi-Fi d'uns 12 km a una freqüència de 5 GHz (d'ús lliure). Per trobar l'emplaçament de l'antena va ser necessari obrir camí al cim d'una muntanya, fins llavors inaccessible, amb un desnivell total de 180 metres. Un total de 280 metres de cable elèctric proporcionen l'electricitat necessària a la instal·lació, que també disposa d'una bateria de reserva en cas que es produeixi un tall en el subministrament. La connexió de dades és a través d'un cable de fibra òptica, per evitar l'impacte visual que suposaria tenir una antena a la teulada de la casa i perquè la longitud màxima d'un cable Ethernet ronda els 100 metres.

Una particularitat de la masia és que es troba aïllada de la xarxa elèctrica i s'autoabasteix amb energia solar, de manera que va ser necessari minimitzar el consum elèctric de tots els components.

Durant l'anàlisi de la instal·lació vam detectar que podíem haver construït un para-llamps sense adonar-nos-en i que, per tant, calia buscar un element de protecció davant de les sobretensions produïdes pels llamps.

En aquesta memòria hi ha especial èmfasi en la part informàtica i de gestió ja que són dos àmbits en els que tinc particular interès, però sense perdre de vista l'objectiu principal: estudiar els efectes d'un llamp a la instal·lació i proposar un prototip per evitar riscos.

Espero que el lector trobi interessant el viatge dels darrers sis mesos, reflectit en aquest document.

3. Introducció

3.1. Objectius del projecte

L'objectiu principal d'aquest Treball Fi de Grau és proposar un dispositiu que pugui evitar les sobretensions causades pels llamps a dins de l'habitatge descrit al prefaci. Es tracta d'estudiar els efectes elèctrics d'una descàrrega a l'antena per tal de dissenyar i construir un prototip que s'adapti a la realitat de la instal·lació existent, aconseguint aïllar la part de l'habitatge i de la muntanya durant una tempesta elèctrica.

A més, és una bona oportunitat per aprofundir en temes d'informàtica i de gestió de projectes, sent aquest últim clau per a l'enginyer industrial. En particular, un altre objectiu del projecte és adaptar *Agile*, una forma de desenvolupar programari i que cada cop més empreses adopten, en un entorn unipersonal.

3.2. Abast del projecte

Aquest projecte conté un estudi bàsic dels efectes d'un llamp a una instal·lació a partir del coneixement adquirit a través de llibres tècnics del camp. Amb aquesta anàlisi es dissenya el prototip amb *SolidWorks* i es construeix.

Es busca que el control del dispositiu, fet amb una Raspberry Pi i programat en *Python*, sigui simple per l'usuari i pugui funcionar de forma autònoma per tal que no sigui necessari intervenir manualment per protegir la instal·lació. Ha de permetre, també, el seu control manual a través d'una pàgina web.

Queda fora de l'abast del treball dissenyar un producte comercialitzable, sinó que es vol que el prototip serveixi de base per, en un futur, construir un dispositiu fiable, barat i eficaç per aïllar dues instal·lacions durant una tempesta elèctrica i evitar, així, la transmissió de sobrevoltatges deguts als llamps.

4. Anàlisi de la problemàtica

4.1. Descàrregues elèctriques atmosfèriques

Un llamp és una descàrrega electrostàtica a través de la qual dues regions de l'atmosfera, o l'atmosfera i el terra, igualen el seu potencial mitjançant una guspira d'entre 5 i 10 km, però que en circumstàncies molt especials pot arribar als 100 km.

L'explicació detallada sobre com es forma un llamp es pot trobar al llibre *Lightning*, de Vladimir Rakov i Martin Uman [38], però en aquest capítol es fa una pinzellada sobre el fenomen natural que ha propiciat aquest treball.



Figura 4.1: llamp núvol-terra. Font: Ron Rev Fenomeno, pixabay.com

Tot el procés es produeix a dins d'un cumulonimbus, un tipus de núvol convectiu de desenvolupament vertical, i és similar al que passa quan es produeix una petita descàrrega al tocar la carcassa del cotxe després de fregar els pantalons amb el seient.

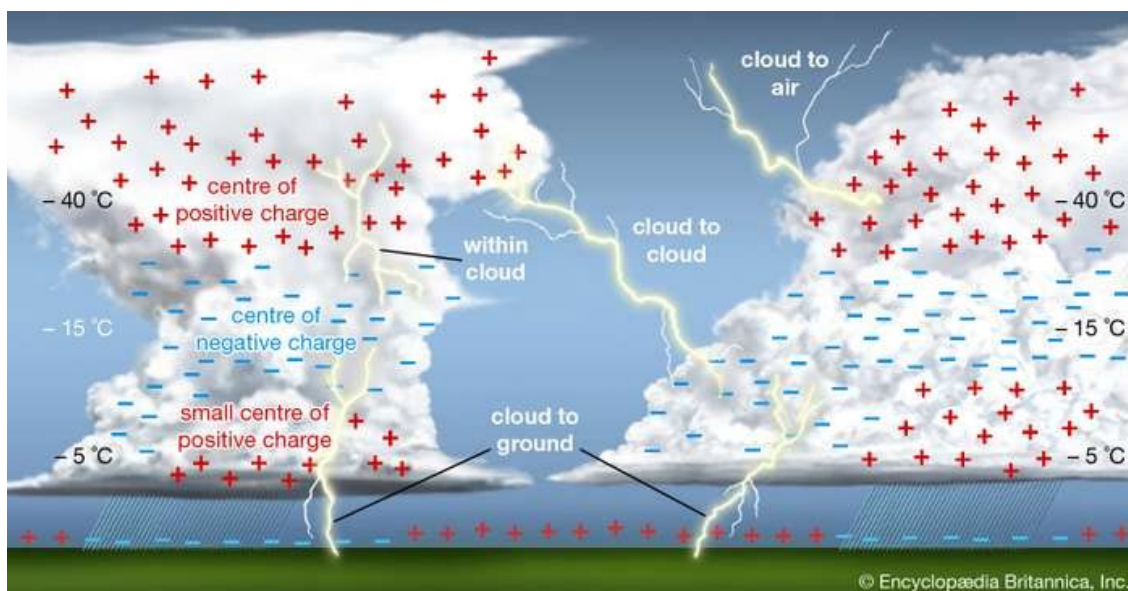


Figura 4.2: distribució de les càrregues en un núvol. Font: Enciclopedia Britannica

A causa de la col·lisió de petits cristalls de gel, suficientment lleugers per ser arrossegats cap amunt pels corrents verticals d'aire, i fragments de calamarsa, més pesats, es transmet la càrrega elèctrica: els cristalls de gel perden electrons mentre que els fragments de calamarsa els guanyen.

Tal i com es pot observar a la Figura 4.2, els cristalls carregats positivament pugen fins al cim del núvol, de manera que aquest queda amb desenes de coulombs de càrrega positiva a la part superior i aproximadament la mateixa en negatiu a la part inferior. A més, la col·lisió entre diferents tipus de precipitació fa que a la part més inferior es pugui crear una petita bossa de càrrega positiva.

De la manera que queden distribuïdes les càrregues es pot diferenciar entre diferents tipus de llamps:

- **Entre núvols:** produït entre la càrrega negativa d'un núvol i la positiva de l'altre.
- **Entre el mateix núvol:** produït entre la càrrega negativa i positiva del mateix núvol.
- **Núvol-terra:** el mostrat a la Figura 4.1, produït entre el núvol i el terra.

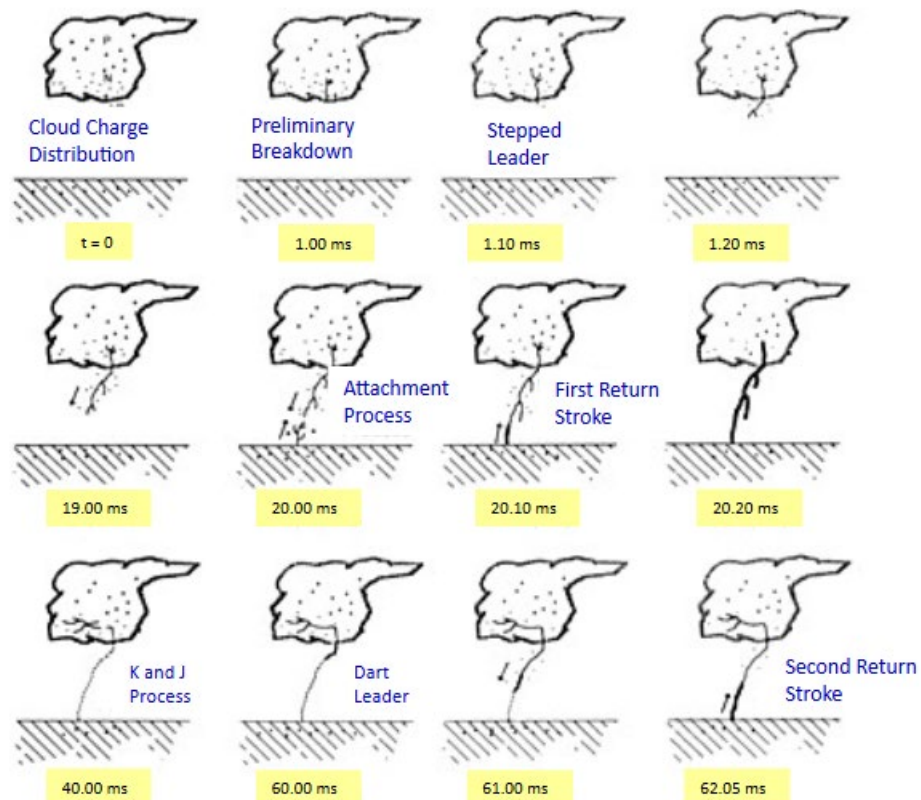


Figura 4.3: procés d'un llamp núvol-terra. Font: Martin Uman (1987)

Aquest treball es centra en els efectes produïts en el darrer cas tal i com es mostra en l'apartat següent, ja que és el que afecta la instal·lació. En l'inici del llamp, la càrrega negativa

contraresta la petita bossa de càrrega positiva inferior però continua el seu descens fins arribar al terra. Aquest moviment no és continu, sinó que es mou de forma discreta en passos de 50 metres d'entre 10 i 50 μ s cada un i amb una velocitat mitjana de 200 km/s.

Quan la capçalera del llamp es troba a una distància suficientment propera del terra, aquest intenta neutralitzar la càrrega negativa amb la creació d'una branca emergent des d'objectes protuberants ben connectats al terra.

Encara falten avenços en el camp de les tempestes elèctriques per entendre-les completament, però aquestes incògnites queden fora de l'abast dels objectius d'aquest treball.

4.2. Efectes d'un llamp sobre la instal·lació

Fer un estudi fidedigne dels efectes d'una descàrrega elèctrica sobre la instal·lació presentada en el prefaci requeriria tot un treball complet, però tal i com s'ha enfocat el projecte no s'ajusta als objectius definits.

S'ha seguit el procediment explicat en el capítol dos del llibre *The Art and Science of Lightning Protection* [56] de Martin A. Uman. Es tracta d'un llibre amb una visió tècnica dels llamps i dels dispositius de protecció i que serveix de guia per qualsevol estudi que s'hagi de fer en aquest camp.

El suport de l'antena WiFi compleix la definició que dona el llibre d'un para-llamps: un objecte protuberant i ben connectat al terra (el mateix suport està en contacte amb el terra i a més està connectat amb cables a piquetes de coure clavades al terra). De fet, segons la introducció feta a l'apartat anterior, aquest suport seria un candidat idoni d'on sortir la branca de càrregues elèctriques positives per enllaçar amb el llamp descendent.

Segons l'autor del llibre esmentat, la major part de la càrrega transmesa a través del llamp aniria directament al terra a través del suport i les piquetes de coure ja que és el camí de menor resistència, però no es pot assegurar que el corrent a través del cable elèctric que enllaça la instal·lació de l'antena amb la casa sigui zero. No es pot conèixer el valor exacte d'aquest corrent i, per tant, se suposa que el 50% de les càrregues degudes al llamp circularan per aquest cable.

Tot i que segurament seria un valor inferior, per tal de dissenyar el dispositiu de protecció de forma eficaç, usar un valor més elevat assegura que la solució sigui robusta. Aquest corrent circulant pel cable provocarà un sobrevoltatge, que a partir dels paràmetres del coure i del llamp és possible calcular.

Com que part del cable elèctric es troba enterrat mentre que altres trams es troben a la

superfície o volant i no forma una línia recta, és complicat estudiar la distribució del sobrevoltatge induït al cable i el seu valor tal i com es mostra en un article presentat a la 2012 *International Conference on Lightning Protection* [15].

Per simplificar, segons Uman, es pot considerar que el sistema és resistiu (ja que posa l'exemple d'un suport connectat a terra i un conductor elèctric llarg). Per tant, el càlcul d'aquest sobrevoltatge és immediat a partir de la llei d'Ohm.

$$V(t) = R I(t) \quad \text{Equació 4.1}$$

El valor de la resistència es pot calcular a partir de la resistivitat del coure ($\rho = 16,8 \text{ n}\Omega \cdot \text{m}$ [35]), la secció del cable ($A = 1 \text{ mm}^2$) i la longitud del cable ($l = 275 \text{ m}$):

$$R = \rho \frac{l}{A} = 4,62 \Omega \quad \text{Equació 4.2}$$

Segons dades extretes de l'estàndard IEC 62305-1 de la *International Electrotechnical Commission* reproduïdes en el llibre, en un cas extrem el corrent màxim d'un llamp és de 200 kA. Ara bé, el corrent pic del 95% dels llamps és de menys de 80 kA i, per tant, es fa servir aquest valor. Tal i com s'ha comentat anteriorment, no tot aquest corrent circularà pel cable:

$$V_{\text{màxim}} = R \cdot 0,5 \cdot I_{\text{màx}} = 4,62 \cdot 0,5 \cdot 80000 = 184800 \text{ V} \quad \text{Equació 4.3}$$

Coneixent el sobrevoltatge produït degut a la circulació de corrent és interessant conèixer la longitud de l'arc elèctric que pot provocar, és a dir, la distància entre dos conductors sense aïllament entre els quals hi ha la diferència de potencial calculada a partir de la qual l'aire condueix i per tant es forma un arc entre ells. Aquest serà el valor usat en el següent capítol per tal de dissenyar el dispositiu.

La tensió de ruptura per l'aire no és lineal i varia segons la pressió i humitat [30]. Considerant que per a una humitat del 60% la tensió de ruptura és de 670 kV m^{-1} segons les dades experimentals de l'article citat, la longitud de l'arc és de 0,27 m.

4.3. Dispositius de protecció existents

Al mercat existeixen diverses solucions que permeten protegir una instal·lació de les sobretensions degudes a les descàrregues elèctriques atmosfèriques. El capítol sis del mateix llibre d'Uman usat en l'apartat anterior conté explicacions d'aquests dispositius.

Un protector de sobretensió transitori, també conegut com a SPD, aconsegueix reduir la sobretensió a un valor residual que difícilment danyarà els components electrònics connectats a la instal·lació.

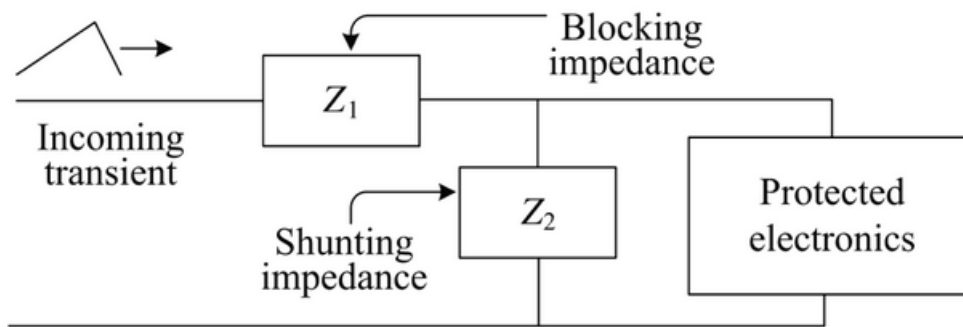


Figura 4.4: esquema de la instal·lació d'un SPD genèric. Font: Standler (2002)

En condicions normals, la impedància Z_1 s'ha d'assimilar a un curtcircuit mentre que la Z_2 , que seria l'SPD, ha de ser suficientment alta per considerar que el circuit està obert per aquesta branca. En canvi, quan hi ha una sobretensió transitòria, la impedància Z_1 l'intenta *bloquejar* i la Z_2 passa a ser un curtcircuit, de manera que no arriba als dispositius connectats.

Per aconseguir aquesta funcionalitat es pot fer servir un circuit *crowbar*, que pot consistir en dos elèctrodes separats per aire, un tub de gas (dos elèctrodes separats per un gas de baixa pressió) o bé circuits electrònics amb trístors, rectificadors i triacs. En el cas del circuit més antic i més barat, dos elèctrodes de carboni separats una fracció de mil·límetre, el temps de resposta és de l'ordre dels microsegons, però té el desavantatge que un cop iniciat l'arc és difícil aturar-lo, danyant el dispositiu.

Una altra alternativa són els fixadors de voltatge com els varistors d'òxid metàl·lic (MOV) o díodes Zener. Tenen l'avantatge de tenir un temps de resposta de l'ordre dels nanosegons i els MOV es poden construir per conduir desenes de milers d'amperes.

Normalment es fan servir dispositius de les dues alternatives en paral·lel, per tenir els avantatges de tots ells i minimitzar el sobrevoltatge que patirà la instal·lació. Com a contrapartida, aquests elements de protecció no asseguren que la tensió residual sigui suficientment petita per no danyar els electrodomèstics i són cars: per exemple, el model A9L16292 d'Schneider Electric té un preu de venda al públic d'uns 400€ [50].

És per aquest motiu que en aquest treball es pretén dissenyar un dispositiu que permeti evitar totalment la transmissió de sobrevoltatge a un preu molt més assequible.

5. Visió global del prototip

En els capítols següents es descriuen de forma detallada tots els components que formen la solució proposada en aquest treball. Aquí es dona una visió global que permet entendre d'una forma més visual i simple com funciona a alt nivell.

El prototip consta de dues parts diferenciades: un mecanisme i un control d'aquest a través d'una Raspberry Pi. Es pot descriure el mecanisme com a un endoll que, a partir d'ordres donades a través del sistema de control, es connecta o desconnecta de forma autònoma.



Figura 5.1: components principals. Font: pròpia

L'usuari pot fer ús d'una pàgina web (veure 8.2.2) per connectar i desconnectar el mecanisme quan cregui convenient o bé deixar que funcioni de forma autònoma. En aquest cas, quan un script que s'executa de forma contínua (veure 8.3) detecta una tempesta propera, es desconnecta automàticament.

Aquesta detecció, tal i com es comenta a l'apartat 7.4, hauria de ser gràcies a un sensor especialitzat. Com que no s'ha pogut provar la seva eficàcia per la falta de tempestes durant el desenvolupament, en aquesta versió s'usen les dades de llamps del Servei Meteorològic de Catalunya.

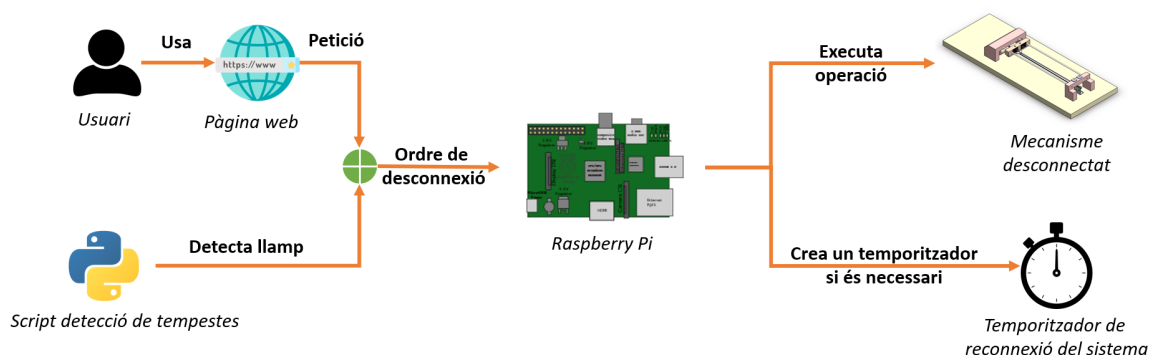


Figura 5.2: procés de desconnexió del sistema. Font: pròpia

Per tal de no requerir de control manual en cap moment, quan el mecanisme es desconnecta de forma autònoma al detectar-se una tempesta, es posa en marxa un temporitzador d'una hora que es reinicialitza cada cop que es detecta un llamp. D'aquesta manera, el mecanisme es connecta automàticament una hora després de la última descàrrega atmosfèrica.

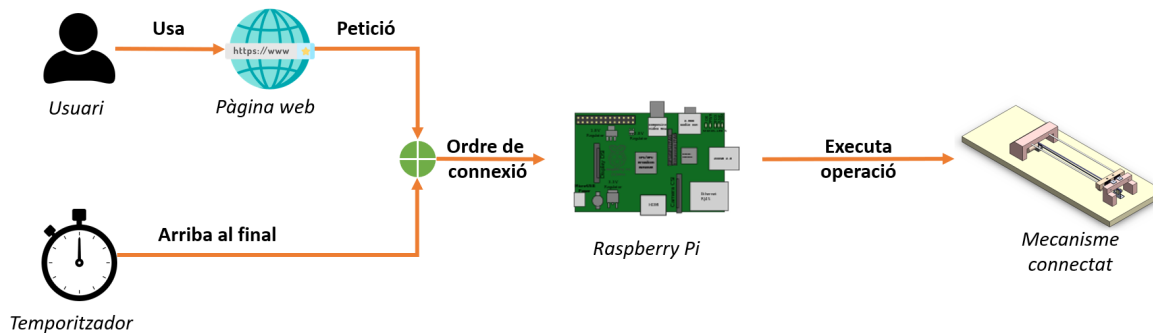


Figura 5.3: procés de connexió. Font: pròpia

El moviment per connectar i desconnectar es fa a través d'un motor que es controla amb un circuit integrat (veure 7.3). Això permet modificar la velocitat a la que es produeix l'acció i escollir el sentit de gir del motor a través de codi *Python*.

Per tal de controlar en tot moment l'estat del mecanisme, es fa servir un sensor de distància ultrasònic (veure 7.2): quan es detecta que s'ha arribat al final de la carrera, s'atura el motor. Per seguretat, s'ha afegit un temporitzador de 45 segons de manera que, si en aquest temps no s'ha finalitzat l'acció, es desconnecta el motor.

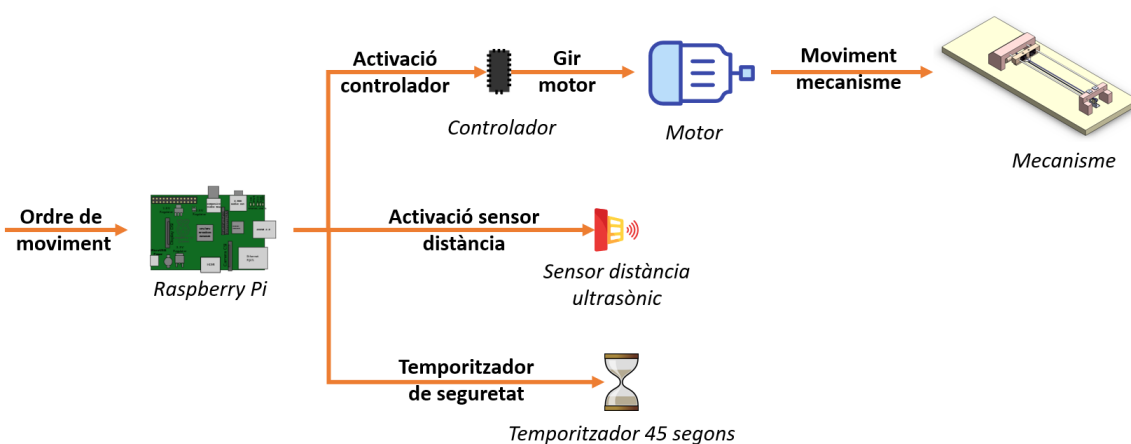


Figura 5.4: inici del moviment del mecanisme. Font: pròpia

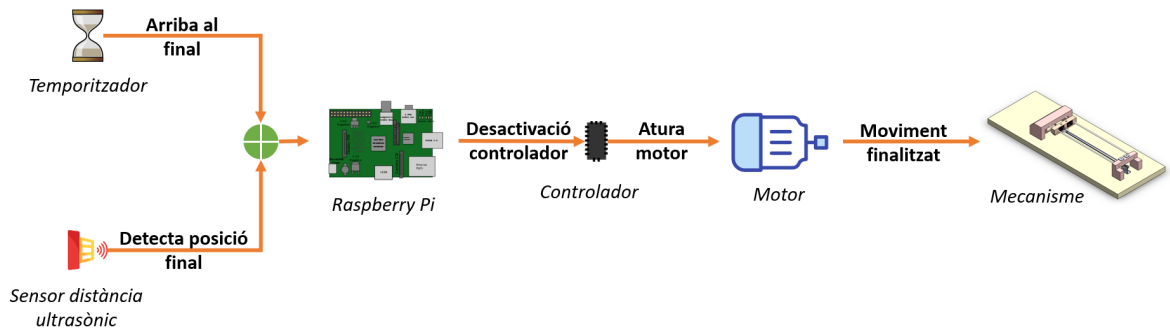


Figura 5.5: final del moviment. Font: pròpia

Totes les accions es reben a través de peticions a una *API* web (veure 8.2.1), programada amb *Python* usant l'entorn de treball *Django*, que conté tota la lògica per controlar el mecanisme. D'aquesta manera, l'script que detecta llamps només cal que faci una petició a l'*API* i aquesta posa en marxa el temporitzador i desconnecta el mecanisme.

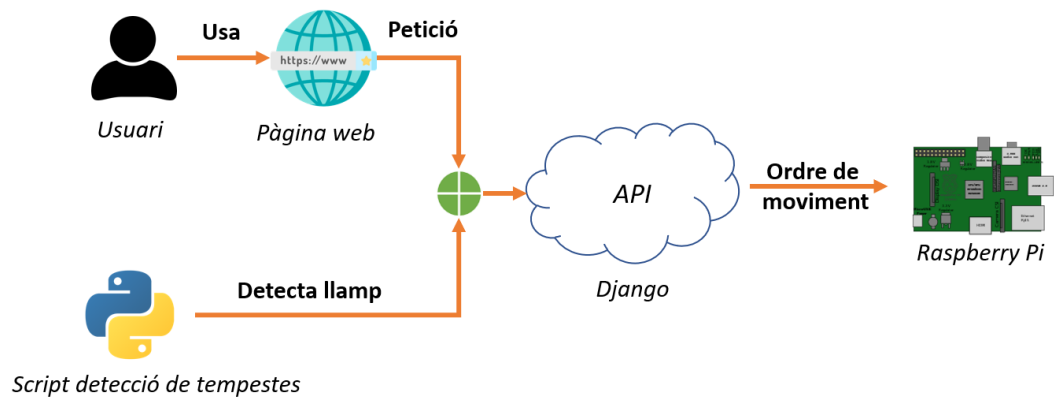


Figura 5.6: peticions a través de l'API. Font: pròpia

6. Disseny i construcció del prototip

Amb l'anàlisi feta al capítol 4 es podria dir que la forma més fàcil d'aconseguir evitar transmetre la sobretensió produïda per un llamp a dins de la masia seria desconnectar físicament la instal·lació elèctrica de l'antena. De fet, el primer dispositiu ideat era una versió molt senzilla: desconnectar el cable de l'endoll estirant un fil enrotllat en un motor, que seria activat remotament quan es cregués convenient.

El desavantatge principal es veu molt clar: un cop desconnectat cal reconnectar la instal·lació de forma manual, cosa que implica estar físicament a la casa. A canvi, s'assegura que la instal·lació elèctrica de l'habitatge no pot patir desperfectes derivats de la caiguda del llamp a la muntanya.

L'inconvenient és massa rellevant i, per tant, una de les condicions que ha de complir el prototip final és que no requereixi de control manual en cap moment. Seguint la informació obtinguda a l'anàlisi del problema, el prototip presentat aquí és un endoll automàtic, assegurant que la distància entre els terminals que uneixen la instal·lació de la casa i de la muntanya sigui superior a la mínima calculada a partir de la qual la sobretensió prevista crearia un arc ja que l'aire deixaria de ser aïllant.

Aquest disseny també té una part negativa, ja que durant una tempesta elèctrica s'interromp el subministrament d'energia a la muntanya. Per evitar que es perdés la connexió a internet s'ha afegit una bateria de reserva que alimenta l'antena durant la tempesta.

Tal i com s'ha comentat, un dels objectius principals és que el cost ha de ser inferior als elements de protecció descrits al capítol anterior i es busca reutilitzar peces per reduir l'impacte ambiental. En els següents apartats es discuteix com han de ser els diferents elements que conformen el dispositiu.

6.1. Elecció del material

Es tenia clar des d'un inici que el material emprat per la construcció del dispositiu havia de ser suficientment aïllant de manera que la tensió de ruptura fos superior al sobrevoltatge calculat al capítol anterior, ja que si no, l'arc elèctric es produiria a través del material.

D'una instal·lació antiga es va poder recuperar una planxa d'espuma de clorur de polivinil (PVC), amb unes característiques mecàniques bones (usat en maquetaria), lleuger i compleix la condició de resistivitat elèctrica. De fet, el recobriment dels cables elèctrics és d'aquest material [17].

Gràcies a les dimensions d'aquesta planxa (500x500x18 mm) s'ha pogut construir tot el prototip reaprofitant material que hauria acabat a la deixalleria i estalviant una quantitat important de diners, ja que tal i com es mostra al pressupost, és car.

Per tal d'unir les diferents peces d'aquest material s'ha fet servir adhesiu específic que té la propietat de *soldar* les parts, aconseguint una resistència molt elevada. A més, al ser PVC d'espuma, es pot tallar fàcilment i no hi ha perill d'esquerdar-lo quan es fan forats.

6.2. Estructura i sistema mòbil

Amb el material escollit s'ha dissenyat l'estructura que permet connectar i desconectar el cable de la muntanya. Pel moviment s'ha aprofitat el motor i corretja d'un escàner en desús, de manera que no ha sigut necessari dissenyar el suport del motor ni la forma de tensar la corretja.

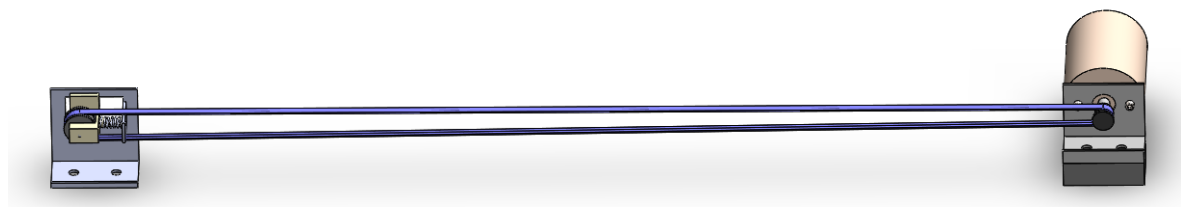


Figura 6.1: sistema motor-corretja-tensor. La geometria real és més complexa, però al ser un sistema reaprofitat únicament s'ha dibuixat una aproximació. Font: pròpia

A canvi, la distància màxima entre els dos terminals ha passat a ser un paràmetre no modificable, que en aquest cas és de 32 cm. Com que és superior a la calculada en el capítol d'anàlisi, el sistema motriu de l'escàner és una bona opció pel treball.

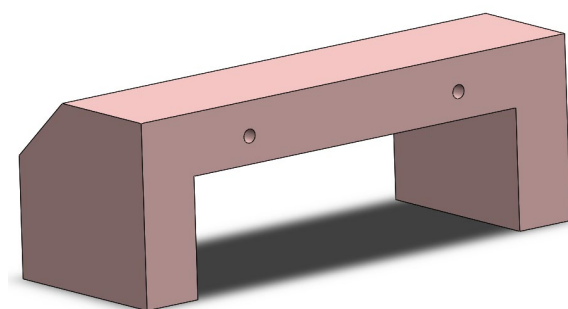


Figura 6.2: extrem del dispositiu amb forma de pont. Al seu interior s'ubicarà el motor i el sensor ultrasònic de distància. Font: pròpia

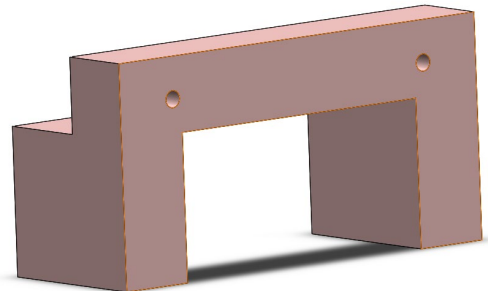


Figura 6.3: extrem del dispositiu amb forma de pont. Al seu interior s'ubicarà el tensor de la corretja i la part vertical contindrà els terminals elèctrics del cable de l'antena. Font: pròpia

Coneixent totes les restriccions s'ha procedit al disseny. Els dos extrems tenen forma de pont per tal de poder situar el sistema motriu a la seva part inferior i en un d'ells se situa el terminal elèctric del cable de l'antena.

La part mòbil, enganxada a la corretja per mitjà d'una pinça feta amb una petita placa de plàstic, es mou a través de dues guies de fibra de vidre i polièster (un material també aïllant, i suficientment rígid i llis per tal que les pèrdues per fregament siguin mínimes) i conté els terminals del cable elèctric connectat a la instal·lació de l'habitatge.

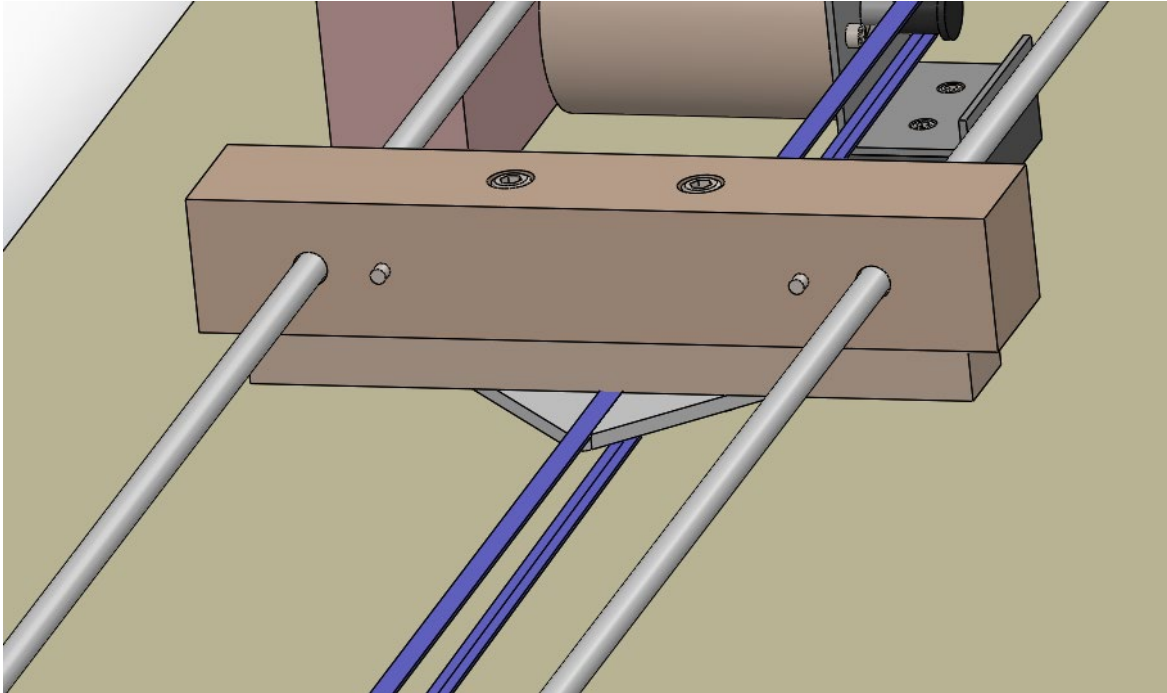


Figura 6.4: part mòbil. Es pot veure la pinça a la seva part inferior pressionant la corretja i l'estructura mòbil. A la part vertical es col·locaran els terminals elèctrics provinents de la instal·lació de la casa. Font: pròpia

L'estructura de la part mòbil és suficientment alta per tal que la seva part inferior toqui la corretja sense necessitat de forçar-la.

Per arribar a la solució proposada, primer es va intentar que el moviment, enlloc de ser transmès a través d'una corretja, fos a través d'una vareta roscada, de manera que una femella fixa a la part mòbil provocaria que aquesta es desplaçés de forma longitudinal. Aquesta idea es va abandonar ja que era molt complicat centrar perfectament la vareta amb l'eix del motor: calia dissenyar una peça a mida.

6.3. Terminals elèctrics

Per trobar la millor forma de connectar el conductor de l'antena ha calgut varies iteracions. En un primer moment es va intentar usar un endoll (connector mascle) pla en moviment que es connectaria a una presa de corrent (connector femella) fixa tal i com mostren els extrems del cable de la següent figura:



Figura 6.5: exemple de connectors plans. Font: amazon.es

Hi havia el problema que calia centrar els dos extrems amb molt bona precisió i qualsevol moviment provocava que el sistema no fos capaç de connectar l'endoll. A més, feia falta un motor potent ja que, tal i com qualsevol pot comprovar, cal fer força per connectar i desconnectar.

És per aquest motiu que s'han dissenyat uns connectors per contacte a partir de material reaprofitat que permeten tenir una tolerància important en el centratge dels dos extrems i no requereixen d'un motor potent.

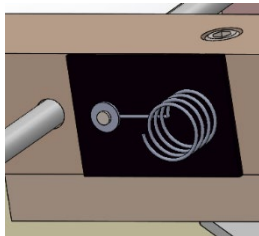
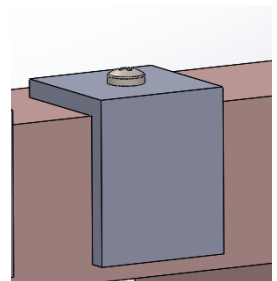


Figura 6.6: terminal elèctric amb molla. Al cargol de la part esquerra és on es connecta el cable de la masia. Font: pròpia

Figura 6.7: terminal elèctric pla. Al cargol superior es connecta el cable provinent de la muntanya. Font: pròpia



Per la part mòbil, que correspon a la instal·lació de l'habitatge, s'ha fet servir una molla estreta d'un porta-piles que es connecta amb el cable a través del terminal mostrat a la Figura 6.8. El fet d'usar molles assegura que el contacte es produirà fàcilment: en cas d'una petita rotació

de la part mòbil, aquella que quedi més propera al connector pla es comprimirà una distància suficient per tal que l'altre terminal faci contacte.



Figura 6.8: exemple de terminal. Font: amazon.es

Per la part fixa, on està connectat el cable de la muntanya, s'ha aprofitat un perfil de 1000x15x1 mm al qual s'ha donat forma de L amb les dimensions adequades i el cable també es connecta a través d'un cargol. Tant la molla com aquestes plaques són d'alumini ja que es tracta d'un bon conductor elèctric (no tant bo com el coure [23]) i de baix pes i preu.

6.4. Ubicació dels components

Tota l'estructura es troba sobre una base de PVC de 630x230x18 mm i s'han repartit els components de manera que el conductor de la muntanya sempre es trobi sempre a almenys 32 cm dels dispositius electrònics i de la instal·lació de la masia.

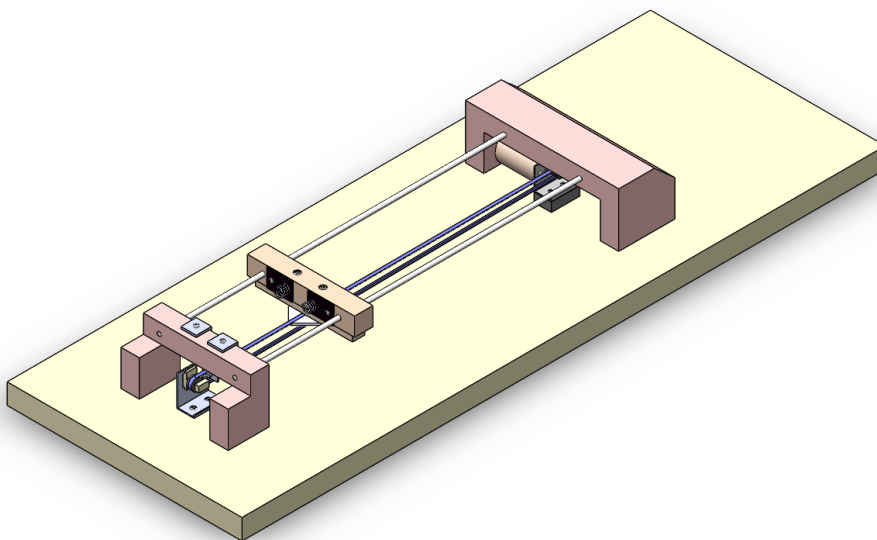


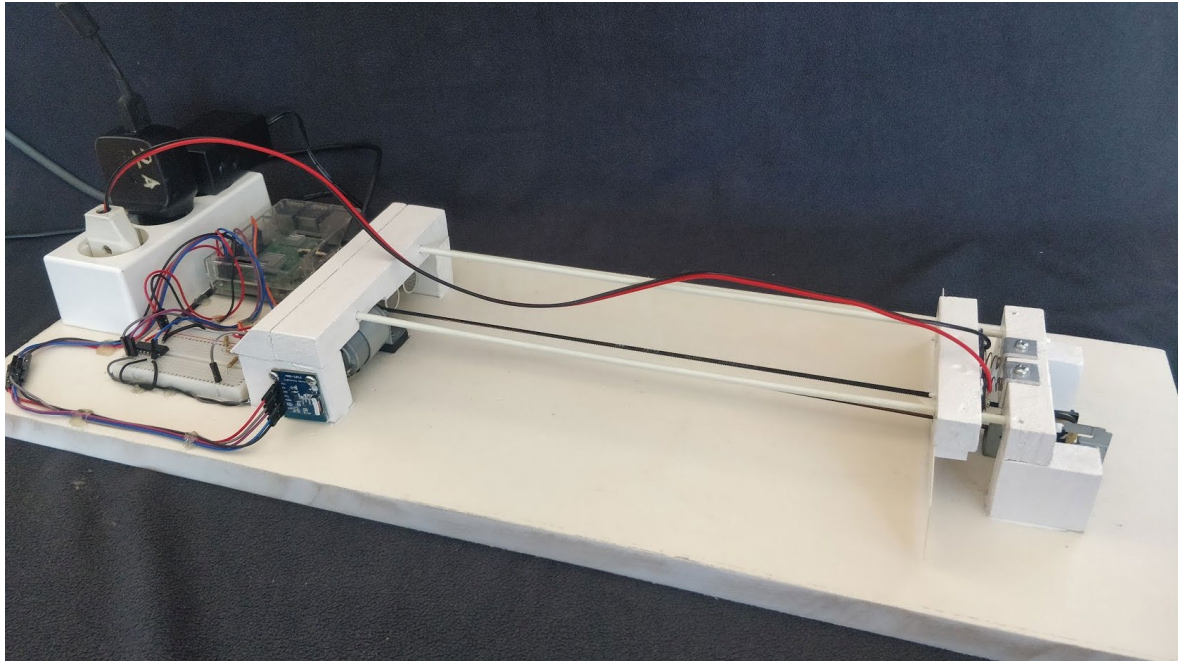
Figura 6.9: prototip dissenyat on es poden veure tots els components descrits en els apartats anteriors. Font: pròpia

En el capítol següent es discuteix quins són els elements necessaris per controlar el dispositiu, però aquí es mencionen ja que s'han col·locat a aquesta base:

- **Raspberry Pi**: el control del dispositiu
- **Protoboard**: per fer les connexions entre els diferents sensors
- Sensor de tempesta **AS3935**
- Sensor ultrasònic de distància **SRF005**: durant la prova del dispositiu s'ha detectat que per tal que fos precís requeria que la superfície a mesurar fos més gran. Per aquest motiu el prototip final incorpora una cartolina a la part mòbil.

A més, s'ha col·locat una base amb disponibilitat per a tres endolls de manera que del prototip només emergissin dos cables: el que alimenta tots els components i el que transporta energia a la instal·lació de l'antena.

A l'annex B es poden trobar diverses imatges que mostren el prototip construït i al D es poden consultar els plànols de totes les parts. En aquest apartat només es mostra la següent fotografia:



*Figura 6.10: prototip construït, on es poden identificar tots els components descrits en aquest capítol.
Font: pròpia*

7. Components electrònics

En aquest capítol s'exposen tots els components electrònics que controlaran el dispositiu mecànic presentat anteriorment. Per començar, es selecciona el dispositiu on es resideix tota la lògica necessària i permet el monitoratge i control remot del mecanisme.

A continuació, es planteja quins sensors i controladors són els més adequats per conèixer la posició de l'endoll i moure el motor. Finalment es presenta el xip que permet detectar tempestes elèctriques.

7.1. Raspberry Pi

Per aquest projecte s'ha decidit usar una Raspberry Pi 3 com a ordinador de placa única i de baix consum per controlar el mecanisme, per la seva versatilitat, facilitat d'ús i programació i pel fet que se'n disposava d'una a l'inici del treball.

És suficientment potent com per allotjar una petita pàgina web i disposa de 40 pins GPIO (*General-purpose input/output*), de manera que es pot comunicar amb sensors digitals de forma directa. A més, està preparat per comunicar-se a través del protocol serial I²C, explicat en l'apartat 7.4.3.

En els següents punts s'exposen les diferents propietats d'aquest ordinador.

7.1.1. Paràmetres elèctrics de la Raspberry Pi

L'energia disponible a la casa on es preveu la instal·lació del mecanisme és molt limitada, de manera que és necessari que sigui de baixa potència per tal de minimitzar l'ús de les bateries carregades amb energia solar.

Així doncs, la primera comprovació que cal fer és, en condicions normals, quin és el consum elèctric teòric de l'ordinador. Segons la documentació disponible a la pàgina web de la Raspberry Pi Foundation, el model usat consumeix entre 700 i 1000 mA [39] i es pot alimentar amb qualsevol font d'alimentació que fixi una diferència de potencial de 5 V.

Aquesta font conté un circuit rectificador per tal de convertir corrent altern a corrent continu. Es pot suposar un rendiment del 80% [57] en aquesta conversió.

En corrent continu no existeix la potència reactiva, de manera que calcular la potència és immediat a partir de les dades mencionades anteriorment. Coneixent, a més, el rendiment (η) de la font d'alimentació, per calcular la potència activa i l'energia consumida durant un temps

(t) de la Raspberry Pi, s'aplica que:

$$P_{consumida} = \frac{VI}{\eta} \quad \text{Equació 7.1}$$

$$E_{consumida,t} = P_{consumida}t \quad \text{Equació 7.2}$$

Per tant, la potència elèctrica consumida prenent un consum mitjà de 850 mA és de:

$$P_{Consumida} = 5,32 \text{ W}$$

$$E_{consumida,dia} = 127,5 \text{ Wh}$$

La instal·lació de generació elèctrica mitjançant energia solar té les següents propietats segons el disseny de l'enginyer de l'empresa *Wattia* [58]:

Propietat	Valor
Capacitat bateries	12,8 kWh
Consum diari previst	10 kWh/dia
Capacitat instal·lació solar	6600 Wp

Taula 7.1: propietats de la instal·lació elèctrica de la masia. Font: Wattia

En la següent figura es pot observar la simulació mensual de la producció elèctrica:

Power production estimate for off-grid PV:

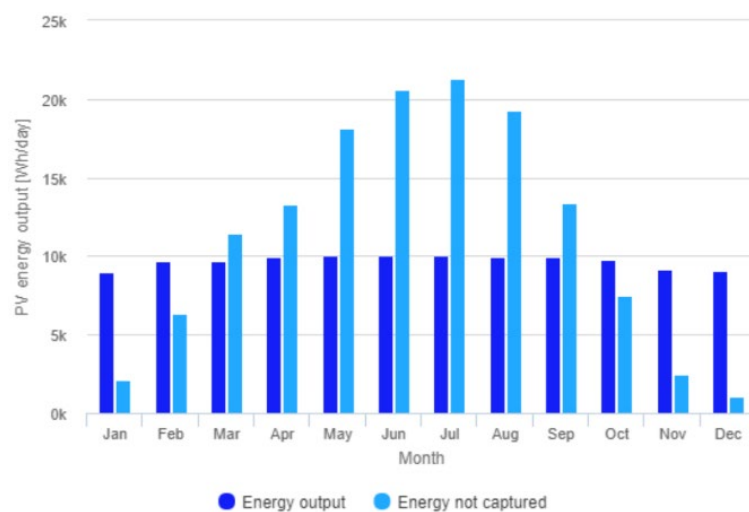


Figura 7.1: estimació de la producció d'energia. Font: Wattia

Tal i com mostra la gràfica, durant els mesos centrals de l'any es captura l'energia suficient per suplir el consum previst i es deixen de produir fins a 20 kWh/dia. Per tant, la Raspberry Pi pot funcionar amb l'excedent, sense interferir amb el consum domèstic.

En canvi, durant l'hivern i la tardor, no es captura energia suficient per suplir el consum diari previst. Segons la simulació es poden capturar uns 9 kWh/dia d'energia, de manera que el consum de l'ordinador correspon a una fracció de:

$$\frac{E_{consumida,dia}}{E_{generada,dia}} \cdot 100 = \frac{127,5}{9000} \cdot 100 = 1,4 \% \quad \text{Equació 7.1}$$

Veient que el consum de la Raspberry Pi és menyspreable davant l'energia generada en un dia d'hivern, es pot concloure que l'elecció d'aquest ordinador és adequada per aquesta instal·lació.

Un altre aspecte important és conèixer si la Raspberry Pi pot alimentar a tots els sensors a través dels pins GPIO: cal assegurar-se que el consum d'aquests no excedeix els límits de la placa. Segons les dades tècniques [8] hi ha les següents limitacions:

- Dels pins d'alimentació de 3,3 V es pot consumir un màxim de 50 mA.
- Dels pins d'alimentació de 5 V es pot consumir com a màxim la diferència entre el corrent màxim de la font d'alimentació (2 A) i el consum real. Considerant un consum real de 850 mA, en total pot circular 1150 mA.
- D'aquells pins que estiguin configurats com a sortida es pot extreure un màxim de 16 mA.
- Els pins configurats com a entrada suporten un potencial màxim de 3,3 V.

En total hi ha tres xips connectats, amb els següents consums màxims:

<i>Dispositiu</i>	<i>Consum màxim (mA)</i>
<i>SRF005 (sensor ultrasònic) [45]</i>	50 mA
<i>L293D (controlador motor) [53]</i>	60 mA
<i>AS3935 (detecció tempestes) [4]</i>	350 µA

Taula 7.2: consum dels sensors connectats. Font: pròpia

Per tant, en total, els xips connectats consumiran un màxim de 110,35 mA del pin de 5 V, de manera que segons les limitacions anteriors, la Raspberry és suficient per alimentar-los.

7.1.2. Característiques físiques

Una de les propietats que fan més interessant l'ús de la Raspberry és la seva mida: 85x56 mm [40], idònia per integrar en màquines i dispositius de l'anomenada casa connectada o casa intel·ligent [21].

No disposa de memòria interna integrada, sinó que el sistema operatiu s'instal·la sobre una SD de manera que és necessari disposar d'un ordinador amb lector de targetes SD per fer la primera instal·lació.

S'alimenta a través d'un connector microUSB i disposa de sortida HDMI i àudio, així com de 4 connectors USB 2.0 i Ethernet de 100 Mb. En el següent esquema es pot veure la disposició d'aquests elements.

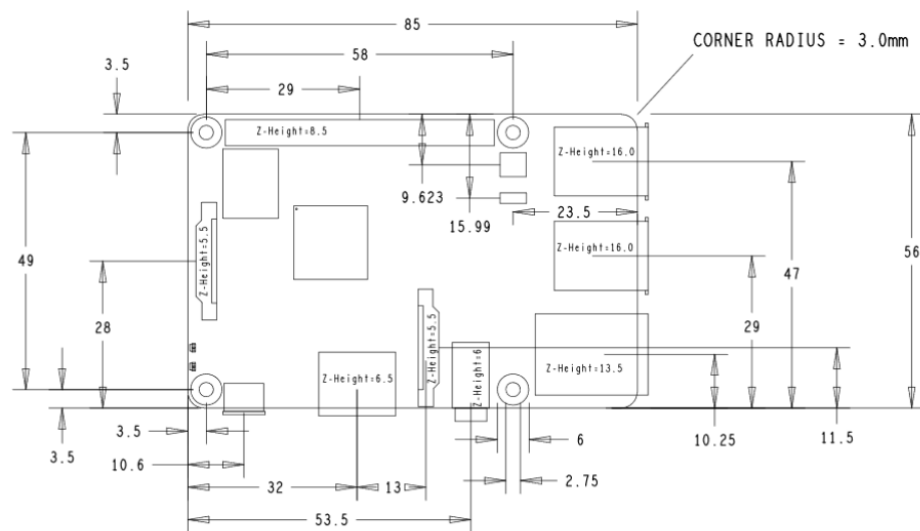


Figura 7.2: esquema de la Raspberry Pi 3B. Font: raspberry.org

Tal i com ja s'ha comentat, s'utilitzen els pins GPIO per comunicar-se amb els diferents sensors. Disposava de dues sortides de 5 V i una de 3,3 V per alimentar els sensors així com diversos pins de potencial 0 V (ground).

També disposa de dos pins especials anomenats *ID EEPROM*, que no es fan servir en aquest treball. Serveixen perquè cert maquinari pugui comunicar-se a través d'un protocol especial per a la Raspberry anomenat HAT [41] (per les seves sigles en anglès, *Hardware Attached on Top*). A través d'aquests dos pins es pot enviar la informació de quins són els GPIO que han d'estar actius, de manera que és molt senzill usar sensors.

La resta de pins, un total de 26, serveixen d'entrades i sortides de senyals digitals. Es pot veure la seva disposició en la figura següent. Dos pins especials són els GPIO2 i GPIO3, ja

que serveixen per comunicar-se a través del protocol I²C. En concret, el GPIO2 és la línia de dades i el GPIO3, el rellotge. L'explicació en més profunditat d'aquest protocol es troba a l'apartat 7.4.3. A més, els GPIO12, GPIO13, GPIO18 i GPIO19 permeten la sortida d'una ona quadrada amb una freqüència màxima de 19,2 MHz.

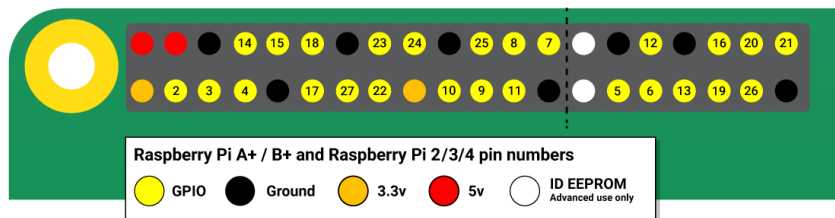


Figura 7.3: disposició dels pins GPIO. Font: raspberrypi.org

7.1.3. Sistema operatiu i llenguatge de programació

Escollit l'ordinador que controlarà el mecanisme es pot decidir quin és el llenguatge de programació que es farà servir.

El sistema operatiu (SO) instal·lat és el desenvolupat per la pròpia Raspberry Pi Foundation: Raspbian. La última versió, Raspbian Buster, va sortir el juliol de 2019 [42] i és basat en Debian 10, igual com altres SO de Linux força utilitzats com l'Ubuntu. Això significa que disposa d'una llibreria extensa de paquets amb instal·lació fàcil a través del gestor *APT* [14].

Com que l'accés és únicament a través de *SSH*, no és necessari instal·lar l'entorn gràfic d'escriptori i així estalviar espai a la targeta SD que fa de disc dur en aquest ordinador. En aquest cas, la SD té una capacitat màxima de 32 GB i queden uns 28 GB lliures per programes i fitxers.

El fet que el SO sigui Linux permet que es pugui usar qualsevol llenguatge de programació suportat, com el *C*, *Java* o *Python*. Per aquest treball s'ha escollit el darrer pels coneixements que es tenien i per la facilitat en el desenvolupament. A més, l'entorn de treball web (*framework*) *Django* [16] està escrit en aquest llenguatge i facilita la creació de pàgines web dinàmiques.

Si bé és cert que *Python* és un llenguatge interpretat, característica que fa que sigui més lent enfront a alternatives compilades com el *Java*, els requisits del projecte i la limitació d'un usuari simultani a la web (ja que serà d'ús intern), són suficientment baixos per menysprear la diferència de velocitat en l'execució.

7.1.4. Alternatives a la Raspberry Pi

Al mercat existeixen microcontroladors que podrien ser adients per aquest mecanisme. Un dels més coneguts i emprats és l'Arduino, que també disposa de pins GPIO i té un consum més baix.

La primera opció que es va explorar va ser programar la lògica en un Arduino Uno, d'una mida semblant a la Raspberry Pi 3B però amb unes característiques molt diferents. La següent taula intenta resumir aquestes diferències dels paràmetres més importants per aquest projecte:

	Arduino Uno [5]	Raspberry Pi 3B
<i>Tipus de dispositiu</i>	Microcontrolador	Ordinador de placa única
<i>Potència elèctrica (W)</i>	0,45	5
<i>Memòria RAM</i>	2 KB	1 GB
<i>Memòria interna</i>	32 KB	No en disposa
<i>Processador</i>	AVR@16 MHz	ARM@1,5 GHz
<i>Número de GPIO</i>	26 (6 analògics)	26 (0 analògics)
<i>Sistema operatiu</i>	Cap	Linux (Raspbian)
<i>Llenguatges de programació</i>	C, C++	Qualsevol suportat en Linux

Taula 7.3: comparació entre Arduino Uno i Raspberry Pi. Font: pròpia

Es pot veure que la finalitat dels dos dispositius és molt diferent. Mentre que la Raspberry Pi està enfocada a ser un petit ordinador generalista, l'Arduino és un microcontrolador programat per un fi concret.

Un dels objectius del projecte és que el mecanisme consumeixi el mínim possible. Així doncs, segons la taula, s'hauria d'usar un Arduino, ja que consumeix molt menys que la Raspberry Pi.

Ara bé, per poder controlar el dispositiu de forma fàcil es fa servir una pàgina web, el servidor de la qual es podria incorporar directament a la Raspberry Pi. Si el moviment es controlés amb l'Arduino, seria necessari igualment disposar d'algun servidor actiu per enviar ordres i controlar l'estat remotament, de manera que al final el consum seria superior.

7.2. Sensor ultrasònic de distància SRF005

Per conèixer la posició de la part mòbil es podria usar un motor pas a pas. Així, comptant el número de passos es podria saber si les dues instal·lacions elèctriques estan connectades o aïllades.

Ara bé, podria ser que de forma externa, de forma voluntària o no, es mogués la part mòbil manualment, provocant que fos impossible per la Raspberry Pi saber si el procés de connexió o desconexió s'ha completat.

Per tant, en aquest projecte s'ha decidit usar un sensor de distància que funciona amb ultrasons. El model escollit és el SRF005 [45], fabricat per PICAXE, que malgrat estar preparat per funcionar amb el microcontrolador dissenyat per la mateixa casa, és compatible amb la Raspberry Pi. Tot i que hi ha multitud de versions al mercat d'aquest sensor amb prestacions similars, es fa servir aquest ja que se'n disposava d'un abans de començar el treball.



Figura 7.4: SRF005. Font: Pròpia

Segons les especificacions tècniques del sensor, permet detectar obstacles a una distància d'entre 3 cm i 3 m d'aquest i a la màxima distància pot diferenciar un objecte de 3 cm de diàmetre. Així, és suficient per l'aplicació que es fa en el prototip.

7.2.1. Funcionament i càlcul de la distància

El sensor disposa de dos transductors, un emissor d'un senyal ultrasònic a una freqüència de 40 KHz i un detector de senyals acústics de la mateixa freqüència. El pols enviat pel primer rebota contra obstacles i és detectat pel segon. Coneixent la velocitat del so, a partir de la diferència de temps entre els dos instants es pot conèixer la distància a la que es troba l'obstacle.

A l'usar una senyal acústica, la mesura no depèn de la quantitat de llum, la presència de fum o brutícia o el color. La única variable del prototip que cal controlar és que la superfície sobre la que es mesurarà la distància sigui rígida [32] per assegurar un bon rebot de l'ona.

Tal i com s'ha comentat, la distància depèn de la velocitat del so, que varia segons el medi i les condicions en les que es troba. A partir de coneixements de Mecànica de Fluids i Termodinàmica es pot deduir que la velocitat de l'ona depèn de la temperatura segons l'equació 7.2. La deducció sencera es pot trobar en obert al llibre *University Physics* [27].

$$v = \sqrt{\frac{\gamma RT}{M}} \quad (m/s) \quad \text{Equació 7.2}$$

On:

Símbol	Significat	Valor per l'aire
γ	Coeficient de dilatació adiabàtica	1,4. Varia amb la temperatura, però en el rang 10-30°C la diferencia és menyspreable [59]
R	Constant universal dels gasos ideals	8,31 J/mol·K
M	Massa molar del medi	0,02897 kg/mol
T	Temperatura	Variable. En graus Kelvin

Taula 7.4: variables que intervenen en l'equació de la velocitat del so segons la temperatura del medi. Font: *University Physics*

Segons dades recollides al llarg d'un any, es pot considerar que la temperatura de l'espai on s'instal·larà el dispositiu oscil·la entre els 10°C a l'hivern i els 30°C a l'estiu. La següent taula mostra la diferència entre les velocitats i les compara prenent com a referència la velocitat a una temperatura de 20°C.

Temperatura (°C)	Velocitat del so (m/s)	Diferència respecte referència (%)
10	338,4	-1,7
20 (referència)	344,3	0
30	350,1	1,7

Taula 7.5: velocitats del so dependent de la temperatura de l'habitació. Font: pròpia

Si el programa que monitoritza el sensor realitza els càlculs a la temperatura de referència, la següent taula mostra com variaria la distància calculada d'un pols que ha rebotat d'una

superfície que es troba a 32 cm del sensor (el màxim al que arriba el prototip):

<i>Temperatura (°C)</i>	<i>Pols rebut (ms)</i>	<i>Distància calculada amb $v_{so}(T=20^{\circ}\text{C})$ (cm)</i>
10	0,95	32,7
20	0,93	32
30	0,91	31,3

Taula 7.6: distància calculada amb velocitat del so a temperatura de referència. Font: pròpia

D'aquests valors es desprèn que durant l'hivern es corre el risc de no connectar completament el dispositiu. Ara bé, la diferència (1,7%, tal i com s'ha vist a la taula anterior) és suficientment petita com per no justificar la compra d'un sensor de temperatura i calcular en cada instant la velocitat del so. En canvi, enlloc d'aturar el motor al moment en què la distància calculada arriba a 32 cm, es manté el motor alimentat durant uns instants extres.

7.2.2. Filtratge de les dades obtingudes

Durant la prova del sensor s'ha vist que és necessari filtrar les dades obtingudes, especialment mentre la peça mòbil està en moviment. La Figura 7.5 mostra la distància calculada si no es filtren les dades (esquerra) o bé si es filtren (dreta):

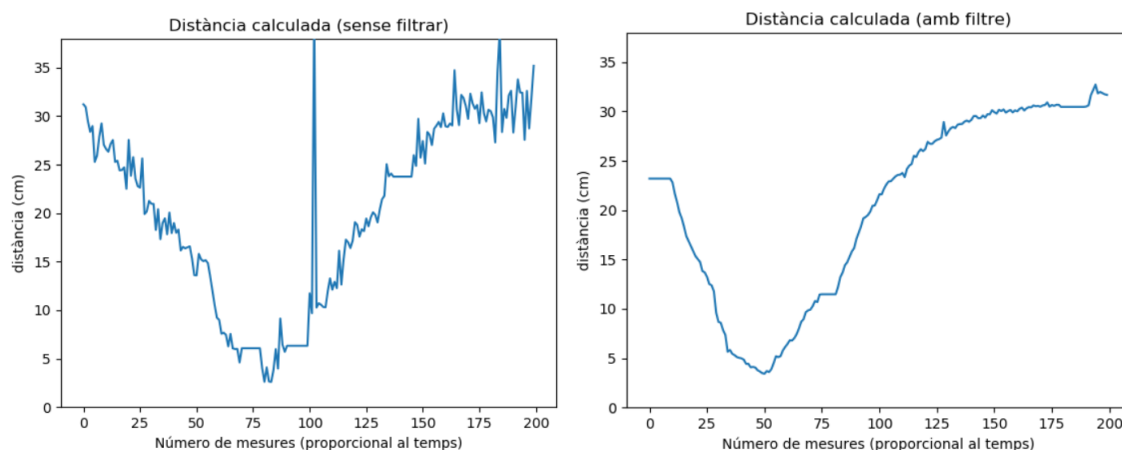


Figura 7.5: comparació gràfica de la distància calculada pel sensor SRF005 amb o sense filtre. Font: pròpia

Hi ha diversos factors que provoquen les variacions observades. Per començar, el SRF005 permet detectar obstacles en tota una zona, de manera que poden haver-hi interferències. En el cas particular d'aquesta versió, el pols ultrasònic de l'emissor emana en un angle de 55°.

La zona de detecció té una forma més complexa. Durant el primer metre des del sensor, aquesta zona s'expandeix amb un angle de 55° i, aproximadament a 2 metres del sensor, la zona de detecció forma un arc d'uns 40° i és màxima, amb una amplada d'1 metre. A partir d'aquesta posició, disminueix fins arribar a una distància de 3 metres.

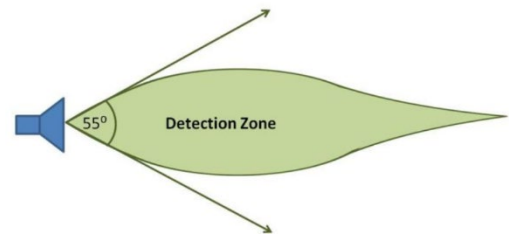


Figura 7.6: zona de detecció del SRF005.

Font: *hobbizine.com*

Com que existeixen altres elements en aquesta zona de detecció en el prototip, és possible rebre interferències. Tot i això, la peça considerablement més gran és la mòbil i la superfície és perpendicular als transductors, de manera que és la que es detectarà més fàcilment.

Una altra font d'interferències són rebots successius. Per evitar que influeixin, cal seguir les especificacions del manual tècnic del sensor: entre pols emesos cal esperar un mínim de 60 ms.

Finalment, cal tenir en compte que es mesura la posició d'una part en moviment, de manera que les vibracions del motor i el mateix moviment poden provocar interferències. A més, la posició rebuda no correspon amb la que la part mòbil es troba en aquell moment. És fàcil modelar aquesta diferència a partir de les equacions de moviment rectilini uniforme ja que interessa tenir bona precisió quan el dispositiu es troba al final de la carrera, és a dir, quan ja no hi ha acceleració.

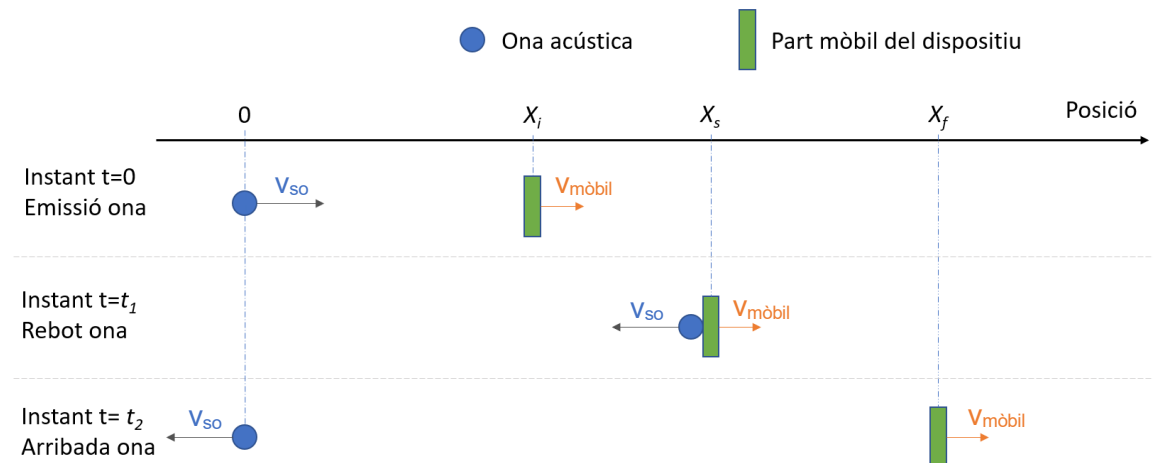


Figura 7.7: modelització del problema per conèixer la posició real de la part mòbil. Font: pròpia

La posició es demana en l'instant $t=0$ quan la part mòbil es troba a la posició X_i , però no és fins a l'instant $t=t_2$ que es rep la posició de la part mòbil, que ja es troba a X_f . El valor rebut correspon, en realitat, a la posició de l'instant $t=t_1$, amb un valor de X_s .

Igualant les equacions del moviment de la part mòbil i de l'ona acústica entre l'emissió i el rebot s'arriba a:

$$X_s = \frac{v_{so}}{v_{so} - v_{mòbil}} X_i \quad \text{Equació 7.4}$$

$$t_1 = \frac{X_i}{v_{so} - v_{mòbil}} \quad \text{Equació 7.3}$$

Si es fa el mateix exercici després del rebot, s'arriba a:

$$X_f = \left(1 + \frac{v_{mòbil}}{v_{so}}\right) X_s \quad \text{Equació 7.6}$$

$$t_2 = \frac{X_c}{v_{so}} + t_1 \quad \text{Equació 7.5}$$

L'equació d' X_f implica que a mesura que la part mòbil s'allunya del sensor, la lectura perd precisió. Considerant que $v_{mòbil}=1$ cm/s i $v_{so}=340$ m/s, al punt més allunyat del sensor (32 cm), la diferència entre la posició real i la retornada pel sensor és de:

$$X_f - X_s = 0,94 \text{ mm}$$

Com que aquesta diferència és inferior a la resolució del sensor, no cal que el programa que el monitoritza compensi per aquest desfasament.

La funció de filtratge s'ha buscat de forma experimental, buscant un equilibri entre la qualitat de les dades i la rapidesa en tenir un valor. La distància actual (x_j) es calcula a partir del valor retornat pel sensor (d), tenint en compte les tres mesures anteriors (x):

$$x_1 = d \quad \text{Equació 7.7}$$

$$x_2 = \frac{d + x_1}{2}$$

$$x_3 = \frac{d + x_1 + x_2}{3}$$

$$x_j = \frac{d}{4} + \sum_{i=1}^3 \frac{x_{j-i}}{4}, \quad j > 3$$

Per tal de donar el valor calculat per bo és necessari que es compleixi:

$$0 \leq d \leq 2,5x_j \quad \text{Equació 7.8}$$

Si d no compleix la restricció anterior no es guarda el valor pel càlcul de la distància en els següents valors retornats pel sensor. El fet d'usar els darrers 3 valors acceptats provoca que la part mòbil no es pugui desplaçar a excessiva velocitat, ja que sinó d sempre es troba a fora dels valors acceptables.

7.2.3. Connexió a la Raspberry Pi i control amb Python

El sensor escollit té dos modes d'operació, però només un d'ells és compatible amb la Raspberry Pi. És l'anomenat *Mode de pins duals* en la documentació, que significa que un pin serveix per enviar un pols (*trigger*) i un segon retorna la distància (*echo*).

S'alimenta des del pin de 5 V i el pin *trigger* es connecta a qualsevol pin GPIO disponible configurat com a sortida. Com que s'alimenta a 5 V, el voltatge obtingut al pin *echo* danyaria el GPIO de la Raspberry Pi, de manera que s'ha fet servir un divisor de tensió amb tres resistències d'1 k Ω per obtenir un voltatge de 3,3 V.

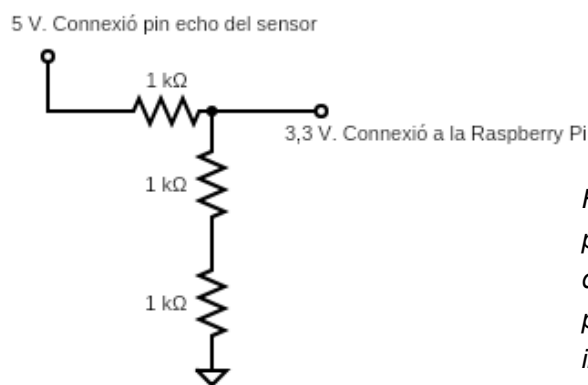


Figura 7.8: divisor de tensió per aconseguir 3,3 V a partir d'una font de 5 V. El càlcul per comprovar-ho és immediat a partir de la llei d'Ohm. Font: pròpia

Usant el programa *piscope* (Figura 7.9) es pot veure l'estat dels pins connectats amb el sensor. S'observa un primer pols de 10 ms (el mínim segons la documentació) que activa el SRF005 i emet l'ona a 40 kHz. Al cap d'uns instants es rep un pols de més llarga duració, que és proporcional a la distància tal i com s'ha comentat anteriorment. La longitud del pols s'ha de dividir per dos, ja que correspon a la distància d'anada i tornada de l'ona.

Per fer més fàcil comunicar-se amb el sensor des del programa desenvolupat en aquest treball, s'ha creat un mòdul de *Python* que conté un objecte amb totes les funcions necessàries per obtenir la distància de la part mòbil. Des de l'arrel del projecte en *Django* (veure el següent capítol), el codi es troba a `lightning.api.devices.srf005.Distance`.

Per tal d'usar-lo cal crear una instància d'aquest objecte i iniciar el sensor, de manera que de forma asíncrona amb el mòdul *threading* va obtenint i filtrant valors de distància. A partir del

mètode `get_distance(self)` es pot obtenir el darrer valor calculat. Finalment, cal aturar el sensor i alliberar els pins per tal que un altre procés pugui usar-los.

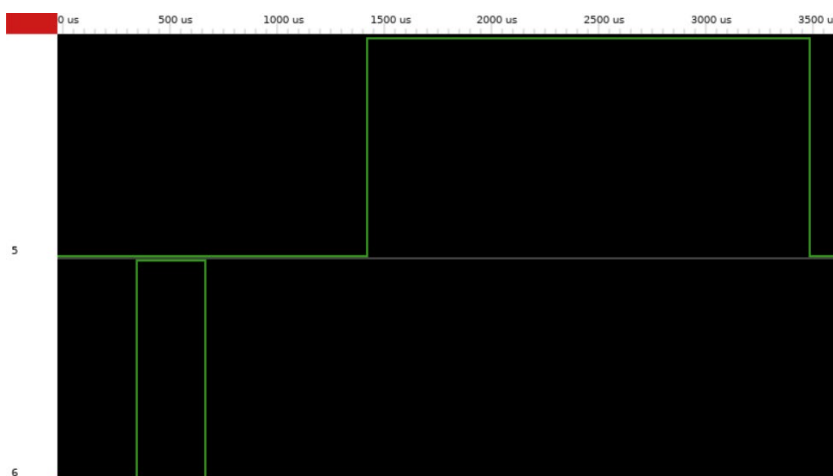


Figura 7.9: captura de pantalla de pìscope. El GPIO6 envia un pols petit per activar l'SRF005 i el GPIO5 retorna un pols la longitud de la qual és proporcional a la distància de l'objecte. Font: pròpia

7.3. Controlador del motor: L293D

La Raspberry Pi no és capaç d'alimentar el motor ja que els pins GPIO no permeten subministrar suficient potència per fer moure el motor i es podrien danyar si s'intentés

Per tant, cal usar un controlador que permeti alimentar el motor amb una font externa i canviar la polaritat de l'alimentació per tal d'aconseguir que giri en ambdós sentits. En aquest projecte s'ha fet servir l'L293D [53], fabricat per Texas Instruments, i és un circuit de mig pont en H.

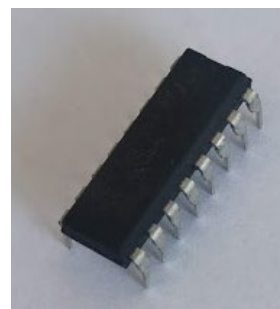


Figura 7.10: L293D. Font: pròpia

En el diagrama de blocs es pot observar com funciona. S'ha usat la connexió representada en la part esquerra, ja que permet modificar el sentit de rotació. Els pins 3 i 6 són les sortides que alimenten el motor a través d'un transistor Darlington [26], activades a partir del bloc lògic 1 i 2, respectivament. La dada més important sobre el funcionament d'aquest transistor és que provoca una caiguda de tensió d'1,8 V.

Com que un motor és una càrrega inductiva, quan s'interromp l'alimentació de forma sobtada l'energia emmagatzemada per les bobines no pot dissipar-se, de manera que pot provocar un alt voltatge que danyaria els transistors. Per tant, aquestes sortides contenen díodes, integrats directament al xip, per protegir-les d'aquests pics tal i com mostra la Figura 7.11.

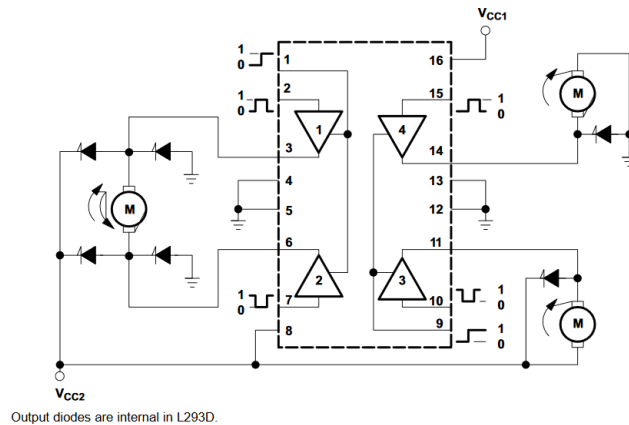


Figura 7.11: diagrama de blocs de l'L293D. Font: ti.com

7.3.1. Modulació per amplada de polsos

Segons les especificacions de la font d'alimentació externa del motor, aquesta pot proporcionar un voltatge de 9 V. Tenint en compte que el transistor provoca una caiguda de tensió de 1,8 V, en realitat el motor s'alimenta a 7,2 V.

Si es prova amb aquestes condicions, es pot observar que el moviment és massa ràpid per ser detectat correctament pel detector de distància ultrasònic. Així, cal buscar una forma de regular la velocitat. Una opció seria posar uns engranatges reductors, però solen ser sorollosos i ocupen força espai.

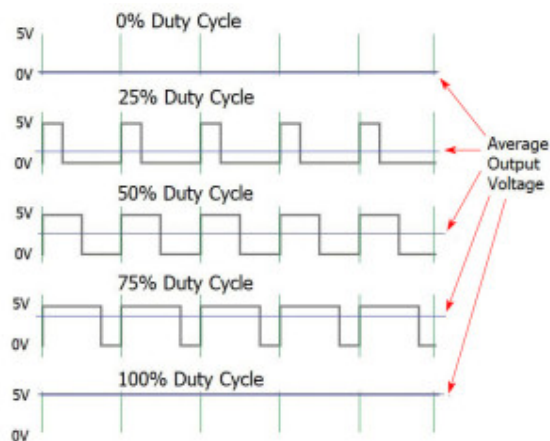


Figura 7.12: ona de polsos d'amplada variable. Font: rudysarduinoprojects.wordpress.com

Així doncs, aprofitant les capacitats de la Raspberry Pi, finalment s'usa la tècnica de modulació per amplada de polsos. L'L293D només s'activa durant una fracció del temps, a partir d'una ona quadrada d'alta freqüència de polsos d'amplada variable. Depenent d'aquesta amplada, el voltatge mitjà que rebrà el motor serà major o menor, variant la velocitat.

A la Figura 7.12 apareix el concepte de cicle de treball, que és la relació entre l'amplada del

pols respecte el total. Si la senyal quadrada es combina amb una font de 5 V a través de l'L293D, en cas que el cicle de treball sigui del 50% el voltatge mitjà serà de 2,5 V, tal i com es pot trobar a partir de l'expressió següent, vàlida per trobar el valor mitjà de qualsevol ona ($f(t)$) a partir del període T :

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt \quad \text{Equació 7.9}$$

De forma empírica s'ha trobat que amb una freqüència de 19 kHz i un cicle de treball del 92%, el motor treballa en unes condicions adequades per tal que el moviment pugui ser registrat pel sensor de distància.

7.3.2. Connexió a la Raspberry Pi i control amb Python

Tal i com s'ha comentat en l'apartat de la Raspberry Pi, aquesta incorpora un nombre limitat de pins GPIO que permeten crear una ona quadrada de polsos d'amplada variable, tal i com la que es necessita.

Com que genera aquesta ona a partir d'un rellotge físic intern, l'ona resultant té tots els polsos idèntics i fa que sigui idoni usar aquests pins enlloc de crear l'ona per programari, ja que no s'aconsegueix la mateixa qualitat.

La connexió és simple: el xip s'alimenta del pin de 5 V de la Raspberry Pi, el pin 1 (activació del controlador) es connecta al pin GPIO amb generador d'ona quadrada, els pins 2 i 7 (control del sentit de la rotació del motor) es connecten a qualsevol pin GPIO disponible configurat com a sortida i els 3 i 6 es connecten als terminals del motor. Finalment, el pin 8 serveix per connectar un dels cables de la font d'alimentació externa.

Tal i com s'ha fet pel sensor de distància, s'ha creat un mòdul de *Python* que permet controlar el motor de forma molt fàcil a través d'un motor. El camí és *lightning.api.devices.l293d.Motor* i el primer pas és inicialitzar una instància de l'objecte. A continuació cal habilitar el motor i iniciar el moviment en el sentit que es vulgui. Finalment s'ha de deshabilitar per alliberar els pins GPIO per un altre procés.

7.4. Sensor de llamps: AS3935

Aquest sensor permet detectar tempestes elèctriques a 40 km i, per tant, decidir si és necessari desconectar la instal·lació elèctrica exterior, de manera que és un component important del projecte. El xip AS3935 és fabricat per l'empresa austríaca AMS AG, però no incorpora els dispositius necessaris per tal que funcioni.

Una de les opcions és comprar la placa MOD-1016 d'*EmbeddedAdventures*, que arriba preparada pel seu ús amb la Raspberry Pi. Malauradament no va arribar segons les especificacions i l'empresa mai va respondre els correus electrònics.

Llegint el manual tècnic de l'AS3935 es va aconseguir fer funcionar la placa MOD-1016 però per la falta de tempestes durant el desenvolupament del prototip, no s'ha pogut provar. Per tant, malgrat que el sensor està connectat a la Raspberry, únicament registra llamps detectats, sense provocar que el dispositiu es desconnecti. Una vegada s'hagi pogut comprovar l'eficàcia detectant llamps, es podrà decidir si és una font fiable.



Figura 7.13: MOD-1016.

Font: pròpia

7.4.1. Funcionament

És probable que si algun cop s'ha escoltat a una ràdio de modulació d'amplitud (AM) durant una tempesta, s'hagi escoltat un soroll que pot recordar el que es fa quan es menja algun aliment cruixent [46]. Això és degut a que una descàrrega elèctrica provoca interferències en les ones electromagnètiques. Si la ràdio se sintonitza a una freqüència de 500 kHz [1] es poden detectar llamps fins a una distància de 340 km amb una bona precisió.

En aquest fet es basa el sensor. La placa MOD-1016 disposa d'una antena formada per una bobina (L1) en paral·lel amb condensadors (C1) amb una freqüència de ressonància de 500 kHz que detecten el soroll i, a través d'un algorisme no públic programat en l'AS3935, permet diferenciar entre descàrregues elèctriques de fins a 40 km de distància i interferències provocades artificialment.

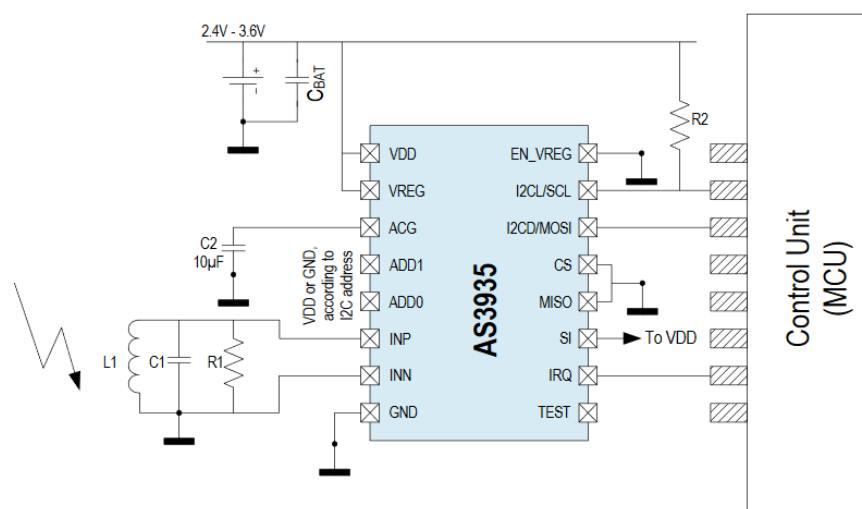


Figura 7.14: diagrama d'aplicació de l'AS3935. Font: ams.com

A més, el xip dona una estimació de la distància a la que s'ha produït la descàrrega i l'energia alliberada. Aquest darrer valor no té sentit físic ja que no és mesurada en Joules, però pot servir per comparar llamps.

La Figura 7.14 mostra el diagrama de tots els components que s'han de connectar quan es comunica a través del protocol I²C i no té el regulador de voltatge activat (veure apartat 7.4.3).

7.4.2. Propietats del sensor

Degut a les variacions inherents en els processos de fabricació dels components, és impossible assegurar que l'antena tingui la freqüència de ressonància adequada. Segons la documentació tècnica, cal que aquesta sigui de $500 \text{ kHz} \pm 3,5\%$. Així, la placa incorpora 15 condensadors de 8 pF que es poden activar individualment amb codi per millorar la precisió.

El xip permet visualitzar la freqüència de ressonància de forma molt senzilla, sense necessitat d'oscil·loscopis. Per tant, cada cop que s'engega el sensor, cal calibrar l'antena afegint o eliminant condensadors. És important que aquest calibratge es faci dins els límits establerts, ja que dos rellotges interns (SRCO a 1,1 MHz i TRCO a 32,768 kHz) basats en oscil·ladors RC [25] depenen de l'antena per funcionar a la freqüència desitjada.

Cada cop que el sensor detecta un esdeveniment avisa a través del pin IRQ, posant-lo a nivell alt. A continuació es pot llegir la memòria del dispositiu, de 51 bytes i l'estructura de la qual es troba disponible a la documentació tècnica [4], que conté què ha detectat (soroll de fons massa elevat, interferència artificial o llamp), així com l'estimació de la distància i de l'energia.

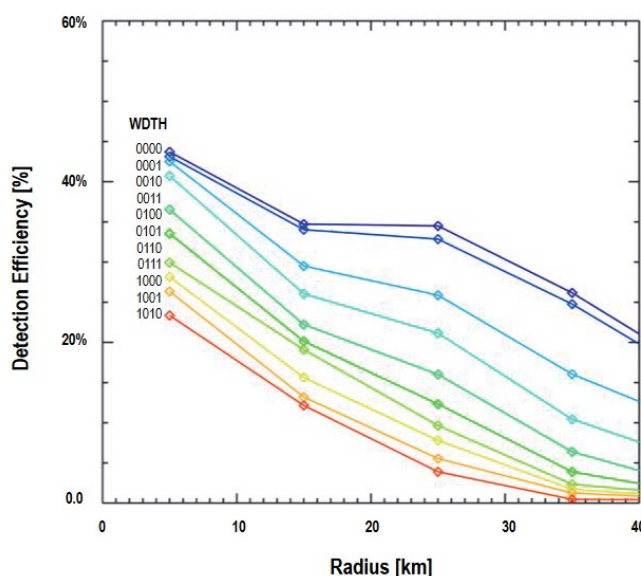


Figura 7.15: eficiència de detecció per diferents valors de WIDTH amb SREJ=0000. Font: ams.com

A més, modificant certs valors de la memòria, permet configurar diferents paràmetres. Per exemple, el número mínim de llamps detectats (registre de nom MIN_NUM_LIGH) abans d'avisar que s'ha detectat un esdeveniment o modificar l'algorisme per millorar l'eficiència al detectar descàrregues elèctriques, canviant la forma del senyal que ha de rebre per decidir que es tracta d'un llamp (SREJ). També permet definir a partir de quin nivell cal que el soroll detectat sigui processat per l'algorisme (WDTH).

Si aquest últim nivell és massa gran, el sensor perd sensibilitat per descàrregues elèctriques llunyanes. La Figura 7.15 mostra que a partir de 25 km, si WDTH és elevat, quasi no es detecten llamps.

7.4.3. Connexió amb la Raspberry Pi

El sensor es pot alimentar a 3,3 V o a 5 V, però si s'escull el darrer, cal activar un regulador de voltatge que consumeix 5 μ A extres. Per tant, com que la Raspberry disposa d'un pin de 3,3 V, s'alimenta a través d'aquest.

El pin IRQ del sensor s'ha de connectar a qualsevol pin GPIO de lectura de dades disponible, ja que el sensor el posa a nivell alt quan ha detectat un esdeveniment.

Per transmetre dades es fa servir el protocol serial I²C [20]. Els pins SDA (dades) i SCL (rellotge) s'han de connectar als pins GPIO 2 i 3 respectivament ja que són els preparats per comunicar-se usant aquest protocol. El sensor té l'adreça 0x03 (programada en la placa, no és possible modificar-la) i la Raspberry Pi inicia la comunicació amb la condició d'inici S (SDA de nivell alt a baix).

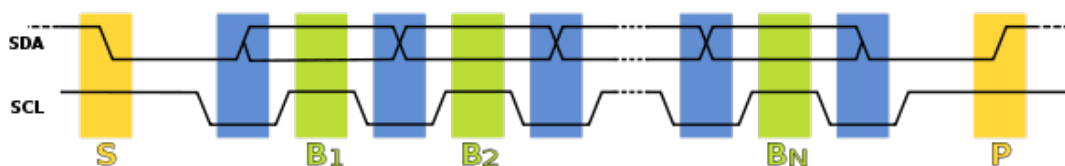


Figura 7.16: protocol I²C. Font: i2c.info

A continuació s'envia l'adreça del dispositiu i el mode (7+1 bits) amb el qual es vol comunicar, l'adreça de la memòria de la qual es vol llegir o escriure (8 bits) i, finalment, hi ha la transmissió dels bits B_i llegits o escrits (normalment un byte, 8 bits). Les dades es transmeten a través del pin SDA a cada pols del rellotge SCL fins que es genera la condició de final P (SDA de nivell baix a alt).

Aquest format és idèntic tant per a la lectura com per l'escriptura. Per saber quin és el mode de comunicació cal mirar el bit menys significatiu enviat amb l'adreça del dispositiu (el bit més a la dreta és el mode, els 7 primers són l'adreça). Si és 0, les dades rebudes pel pin SDA

s'escriuran, mentre que si és 1, el sensor retornarà el byte que hi ha a l'adreça especificada.

Per exemple, si s'envia la següent seqüència de bits entre *S* i *P*:

00000110	00001000	11010011	
Adreça dispositiu	Mode escriptura	Adreça registre	Informació a escriure

S'escriurà al registre amb adreça 0x08 del dispositiu amb adreça 0x03 el byte 11010011.

7.4.4. Programació amb Python

Tal i com s'ha fet amb la resta de sensors i controladors, s'ha creat un mòdul de *Python* que permet comunicar-se amb el sensor de forma molt fàcil. Es troba a *lightning.api.devices.as3935.AS3935* i s'ha publicat al gestor de paquets de *Python PyPi* (<https://pypi.org/project/as3935/>) per tal que pugui ser usat per qualsevol que també tingui aquesta placa.

De la mateixa manera que la resta de mòduls, es tracta d'un objecte amb diferents mètodes que permeten realitzar totes les operacions descrites en el manual de l'AS3935. D'aquesta manera, no és necessari recordar en quina posició es troben els bits que es volen modificar per aconseguir la configuració desitjada. La documentació de totes les funcions es pot trobar en el mateix fitxer o a l'adreça mencionada anteriorment de *PyPi*.

8. Sistema informàtic

En aquest capítol s'explica quin és el sistema de control que s'ha programat i com és el producte final.

Tal i com s'ha comentat durant l'elecció de la Raspberry Pi, el llenguatge de programació usat és el *Python*, ja que té un rendiment suficient per aquesta aplicació en un ordinador de placa única. De fet, aplicacions d'ús quotidià com Instagram, Google o Spotify [9] fan servir aquest llenguatge per les seves aplicacions i, en el cas de la primera, utilitzen el mateix entorn de treball web que en aquest treball.

8.1. Components principals

8.1.1. Control del GPIO: *pigpio*

En el capítol anterior s'han presentat tots els sensors i controladors usats i s'ha comentat que per a cada un d'ells s'ha desenvolupat un mòdul per a poder-hi interactuar de forma simple en la lògica de l'aplicació.

Un dels paquets més usats per controlar l'estat dels pins GPIO és el *RPI.GPIO* [12], però no és suficientment complet. Per començar, no permet usar la modulació d'amplitud d'ona (per l'L293D) creada per maquinari ni suporta el protocol I²C (per l'AS3935). A més, necessita permisos de superusuari per executar els scripts però no és segur donar aquest tipus de permís a una aplicació que serà accessible des de l'exterior a través de web.

Per tant, s'ha fet servir la llibreria *pigpio* [2], que consta de dos components. Per un cantó hi ha el dimoni [47], escrit en C, que es tracta d'un programa que s'executa en segon pla de forma contínua amb permisos de superusuari i que és l'encarregat de modificar el nivell dels pins GPIO configurats com a sortida i llegir aquells configurats com a entrada.

Per l'altre, hi ha el mòdul de *Python*, que permet comunicar-se amb el dimoni i ordenar el canvi d'estat o lectura dels pins. La documentació de totes les funcionalitats es troba a la pàgina web de la llibreria però, a grans trets, en aquest treball s'han usat els mètodes *set_mode*, *read*, *write*, *callback*, *hardware_pwm* i aquells relacionats amb la comunicació per I²C.

Els noms dels mètodes permeten identificar de forma clara què fan. En el cas de *callback*, permet definir una acció quan es detecta un canvi d'estat en un pin GPIO configurat com a entrada. A *hardware_pwm* es configura la modulació d'amplitud d'ona generada pel rellotge físic de la Raspberry Pi.

8.1.2. Django

El *Django* [16] es un entorn de treball web suficientment potent perquè empreses com Instagram l'utilitzin per a les seves aplicacions. Permet crear una pàgina web d'una forma ràpida i senzilla, evitant reescriure codi que és comú en moltes webs.

El codi, una vegada s'ha acabat, s'executa al darrera d'un servidor web *Apache* [54] que també és l'encarregat de subministrar imatges estàtiques com les que es poden veure a la pàgina web creada en aquest treball.

Guarda les dades en bases de dades creades a partir de models, de manera que no és necessari escriure codi SQL. Només cal especificar les columnes necessàries i la configuració de la taula, com ara l'ordre per defecte dels elements. Tal i com es pot veure en la següent figura, també permet crear relacions entre taules de forma molt senzilla amb el camp *ForeignKey*.

```
class Forecast(models.Model):
    """
    A model to save the forecast data from meteo.cat. Rain is in % and sky refers to a Sky object
    """
    date = models.DateField(auto_now_add=False)
    sky = models.ForeignKey(Sky, on_delete=models.CASCADE, to_field="sky_value", default=2)
    temp_max = models.FloatField()
    temp_min = models.FloatField()
    rain = models.FloatField()

    class Meta:
        """
        To set the order, from new to old
        """
        ordering = ['-date']
        get_latest_by = ['date']
```

Figura 8.1: exemple de model. Font: pròpia

Tota la lògica queda englobada en vistes basades en funcions o classes, que processen una petició i retornen un contingut, com ara text simple o *HTML* (HyperText Markup Language) creat a partir d'una plantilla per tal que pugui ser visualitzat per un navegador web.

Quan el servidor rep una petició, el *Django* resol la URL. És a dir, redirigeix la petició a la vista adequada segons el codi ubicat en el fitxer *urls.py*. La resolució es fa a partir de comparar la URL de l'usuari amb un seguit de patrons que es troba en el fitxer esmentat.

A continuació, la vista executa tota la lògica, que pot contenir peticions a la base de dades a través dels models i retorna el contingut. El següent diagrama mostra aquest procés de forma

clara. Hi intervenen altres components, com ara la comunicació entre *Apache* i *Django*, però per simplicitat no s'afegeix.

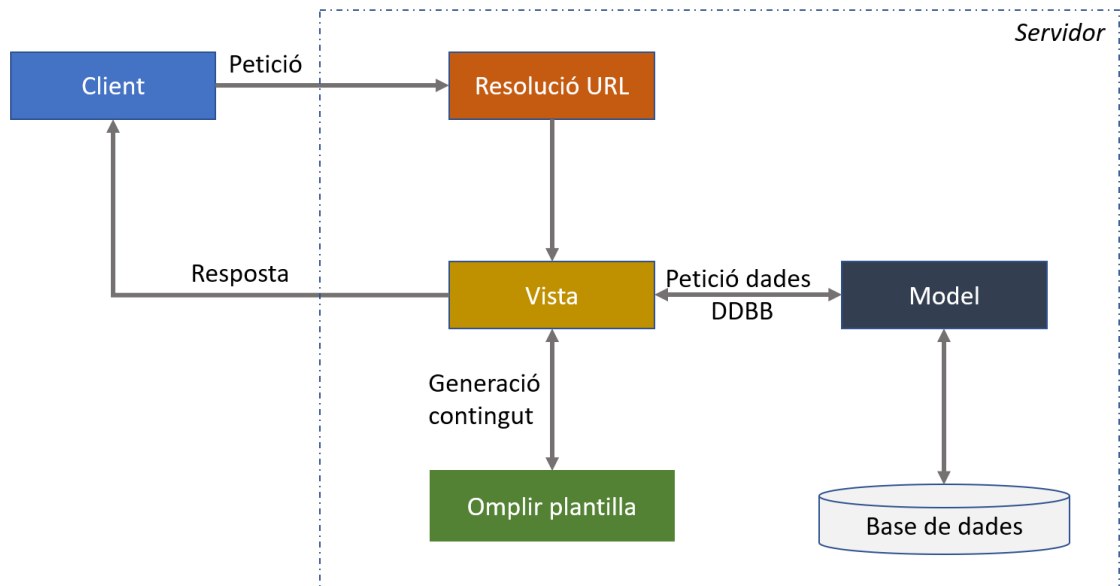


Figura 8.2: procés d'una petició en Django. Font: pròpia

El *Django* funciona a partir d'aplicacions dins d'un projecte, de manera que es poden agrupar funcionalitats en una aplicació per poder ser reutilitzades en diferents projectes. El projecte són totes les funcionalitats de les diferents aplicacions que es poden consultar a través de la web.

8.1.2.1. Django REST Framework

Una API, per les seves sigles angleses d'*Application Programming Interface*, és un protocol de comunicació entre programes [34] que defineix com els diferents components han d'interaccionar.

En el cas d'aquest projecte s'ha programat una API Web, és a dir, s'exposa les rutes d'accés a la informació a través d'URLs (*Uniform Resource Locator*) i funciona a sobre del protocol *HTTP* (*HyperText Transfer Protocol*). Dit d'una altra manera, la comunicació es fa a través d'adreces web, que en aquest cas únicament seran accessibles des de la xarxa local.

Per passar informació s'utilitza el format *JSON* [11] (*JavaScript Object Notation*) que té una estructura molt semblant a un diccionari de *Python* amb parelles atributs-valors. Els valors poden ser de tipus cadena de caràcters, enter, decimal, booleà o llistes. La següent figura mostra un exemple de *JSON*, que, com es pot veure, la seva lectura és immediata per a un humà:

```
{
  "dies_prediccions": 2,
  "ubicacio": "CATBCN08028",
  "prediccions": [{
    "dia": "25/09/2020",
    "temperatura": 29.5,
    "humitat": 80,
    "pluja": true
  },
  {
    "dia": "26/09/2020",
    "temperatura": 31.5,
    "humitat": 75,
    "pluja": false
  }
  ]
}
```

Figura 8.3: exemple de JSON. Font: pròpia

Hi ha diferents estils d'APIs web i s'ha escollit l'arquitectura *REST* (*REpresentational State Transfer*), molt usada recentment. La ruta d'accés informa al servidor quin és el recurs al qual es vol accedir i la operació es coneix a partir del verb *HTTP*: *GET* (demandar un recurs), *POST* (afegir el recurs a la base de dades del servidor), *PUT* (modificar un recurs), *DELETE* (esborrar un recurs). Per exemple, si es fa una petició *GET* a l'URL <https://api.exemple.com/prediccio/CATBCN08028>, el servidor retornaria un JSON com el mostrat a la figura anterior, que correspon a la representació del recurs "predicció a CATBCN08028".

Per tal de ser considerada una *API REST*, cal que es compleixin les següents condicions [7]:

1. La petició ha d'incloure l'identificador del recurs
2. Cada petició ha d'incloure suficient informació per tal que el servidor l'entengui completament. De fet, la comunicació és *stateless*: el servidor no recorda cap informació del client entre peticions consecutives.
3. El client pot interactuar amb el servidor a partir de *hypermedia* proporcionat en les respostes rebudes, és a dir, inclou enllaços per interactuar amb les dades.
4. El client i el servidor son entitats separades. El segon espera que el primer faci una petició.
5. Entre el client i el servidor poden existir capes (de seguretat, de caché, etc), però el resultat final no ha de variar segons aquestes.
6. Les dades s'han de poder guardar en caché per tal de minimitzar el número de peticions.

La implantació d'aquesta arquitectura per visualitzar les dades guardades en la base de dades

és immediata, ja que es compleixen totes les condicions. Ara bé, tal i com es veu a l'apartat de l'aplicació API, també s'han afegit rutes que controlen el motor i el sensor ultrasònic. En aquest cas, el recurs no són les dades sinó el resultat de l'operació que du a terme el mecanisme.

Una vegada vista la utilitat d'una *REST API*, s'ha buscat la forma d'implementar-la en *Django*. El *Django REST Framework (DRF)* [18] és un mòdul que es pot acoblar al *Django* i habilita la creació d'*APIs* d'una forma molt senzilla, sense necessitat de programar des de la base mètodes per crear i llegir el format *JSON*, controlar-ne l'accés, etc.

Inclou, per defecte, una pàgina web on es poden visualitzar totes les rutes possibles de l'*API* i navegar a través d'elles, fent totes les accions *HTTP* mencionades anteriorment de forma intuïtiva i sense necessitat de fer servir eines amb la línia de comanda com el *CURL* [13].

La serialització, procés de convertir un objecte de *Python* en *JSON*, també és directe. Només cal definir de quin model s'extreuen les dades i quins són els camps que ha de contenir la resposta. Funciona idènticament si es llegeix un *JSON*.

```
class LightningStrikeSerializer(serializers.ModelSerializer):  
    """  
    A serializer with all the fields from the LightningStrike model  
    """  
    class Meta:  
        model = LightningStrike  
        fields = ['id', 'date', 'source', 'distance', 'max_current', 'cloud_earth', 'latitude', 'longitude']
```

Figura 8.4: exemple de serialitzador. Font: pròpia

Inclou l'anomenat *ViewSet*, un tipus de vista de *Django*, que conté tota la lògica necessària per realitzar les operacions habituals. Així, en molts casos crear una ruta d'accés és immediat i no cal escriure cap línia de codi addicional.

L'últim recurs útil i que es fa servir en aquest treball són els *Routers*, que creen totes les rutes de l'*API* a partir dels *ViewSet* creats.

Com es pot veure, usar *DRF* és molt útil ja que permet minimitzar el codi escrit i facilita tenir una *API* llesta en poc temps.

8.1.2.2. Aplicacions

S'han desenvolupat dues aplicacions separades. Per una banda hi ha l'*API* que funciona a partir de *DRF*, i per l'altra hi ha la pàgina web.

Això és possible ja que l'*API* no depèn de la pàgina web i podria ser que en algun moment fos interessant deshabilitar l'accés a la pàgina, mantenint l'*API* en funcionament. Per exemple, en

actualitzacions futures de la pàgina, com ara afegir funcionalitats o modificar el disseny.

A grans trets, l'*API* permet controlar el mecanisme i visualitzar dades fent peticions *HTTP* a través d'una *URL* i retorna text en un format concret per conèixer el resultat. La pàgina web fa servir l'*API* per ajudar a l'usuari a controlar el dispositiu d'una forma més visual i intuïtiva. Les explicacions més detallades es troben en un apartat posterior.

8.1.3. Base de dades: PostgreSQL

Per tal d'emmagatzemar dades com ara els llamps detectats, l'estat més recent del mecanisme o la previsió del temps s'usa una base de dades, actualitzada a través de *Django*. Per defecte, s'utilitza el tipus *SQLite* que té la particularitat de ser un sol fitxer i no necessita de servidor. Tot i això, durant el desenvolupament del treball s'ha vist que aquest format era insuficient per l'aplicació i per tant s'ha migrat a *PostgreSQL* [55].

Treballa segons un model de client-servidor, de manera que el *Django* (el client) rep les dades a partir de la resposta del servidor, que pot estar ubicat en qualsevol dispositiu connectat a la xarxa. Per aquest treball, el servidor es troba en la mateixa Raspberry Pi.

L'empresa TablePlus, desenvolupadora d'eines amb interfície gràfica per treballar amb base de dades, té disponible en línia una comparació [52] entre *SQLite* i *PostgreSQL*. En aquesta memòria es reproduïx aquells punts més rellevants.

	PostgreSQL	SQLite
<i>Tipus de dades</i>	Qualsevol tipus	BLOB, NULL, INTEGER, TEXT, REAL
<i>Mida del programari</i>	100 MB	500 kB
<i>Portabilitat</i>	Cal migrar totes les taules a un nou servidor	Només cal copiar el fitxer de la base de dades
<i>Accés multiusuari</i>	Sí	Lectura: Sí Escriptura: No
<i>Velocitat</i>	Al necessitar un servidor, amb la Raspberry Pi es pot notar baix rendiment	Accés molt ràpid

Taula 8.1: comparació entre PostgreSQL i SQLite. Font: TablePlus

A priori, sembla que *SQLite* hauria de ser suficient, però l'existència de tasques asíncrones fa necessari tenir accés multiusuari. A part, podria ser que en un futur s'afegissin altres funcionalitats a la Raspberry Pi i podria ser útil tenir una base de dades com *PostgreSQL* ja funcionant.

El següent diagrama mostra com és la lectura i escriptura en el model client-servidor.

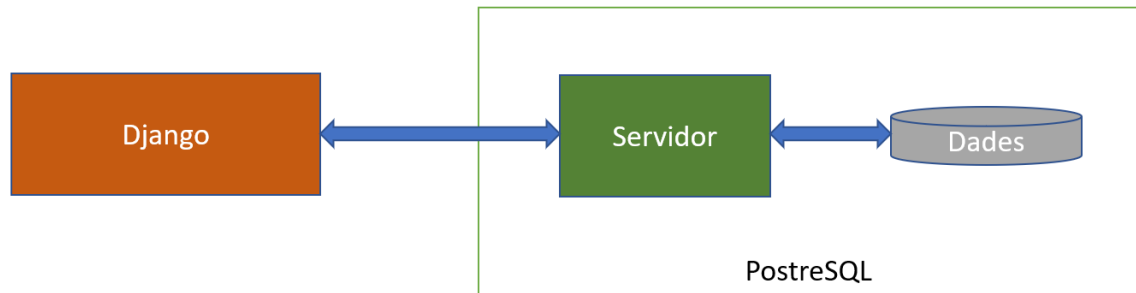


Figura 8.5: comunicació entre PostgreSQL i Django. Font: pròpia

Hi ha altres alternatives a *PostgreSQL*, com ara el *MySQL* o *MariaDB*. Les funcionalitats de totes elles són similars i no hi ha una gran diferència de rendiment. Al final s'ha decidit per la primera perquè no s'hi havia treballat mai i és interessant tocar eines diferents.

Tots els tipus de bases de dades tenen en comú la mateixa llengua per visualitzar i afegir dades: l'*SQL*. El més important, però, és que siguin compatibles amb *Django* ja que és el que gestionarà la base de dades. Aquest és un dels avantatges d'usar-lo, ja que el programador només necessita escriure codi en *Python* per afegir o llegir dades.

Totes les taules necessàries són creades pel *Django* de forma automàtica a través del seu script d'administració. A part de les creades a partir dels models (com ara la taula que guarda tots els llamps detectats), n'hi ha de paràmetres de configuració, dels usuaris (com l'administrador, que permet modificar els elements de la base de dades des d'una interfície web) o de les tasques programades que s'han d'executar en un futur proper.

8.1.4. Gestió de tasques asíncrones: *Celery* i *Redis*

Algunes funcionalitats del mecanisme triguen força estona a completar-se, com per exemple la connexió o desconexió del sistema elèctric, que necessita uns 30 segons. Si un usuari fa una petició a l'*API* per desconectar no voldrà esperar aquest temps amb la pàgina carregant per rebre la confirmació que s'ha realitzat l'acció.

És per aquest motiu que cal buscar una manera de poder executar tasques de forma asíncrona en segon pla, de manera que l'*API* pugui retornar a l'usuari un codi de seguiment del procés de la tasca de forma immediata.

S'ha escollit el *Celery* [51], ja que és integrable amb *Django* i també és usat per empreses amb un volum important de trànsit, de manera que es pot considerar que és fiable. Es tracta d'un dimoni que gestiona una cua de tasques que executa de forma paral·lela fins a un màxim definit en la configuració. L'ordre amb el qual executa les tasques és per l'ordre d'arribada, tot i que en aplicacions més complexes es poden definir prioritats.

Per poder comunicar-se amb el dimoni es fa servir un missatger, que en el cas d'aquest treball és el *Redis* [44]. Quan es vol executar una tasca asíncrona s'envia la petició a través del *Redis* i, a continuació, el *Celery* retorna un número identificador de la tasca per tal de poder-ne seguir l'execució.

Una particularitat del *Celery* és que permet definir tasques que s'executin en instants concrets del dia, cosa que és d'utilitat en aquest projecte per descarregar la previsió meteorològica o els llamps de la darrera hora. A més, també permet definir aquesta planificació en segons des del moment que es rep la tasca: per exemple, quan es desconnecta el sistema, es programa una tasca perquè al cap d'una hora es reconnecti el dispositiu.

La Figura 8.6 mostra com el *Django* fa peticions a través del *Redis* al *Celery*. Es crea una cua ordenada (tasques 4-7) segons l'ordre d'arribada i el *Celery* executa tres tasques (1-3) en paral·lel. Quan la primera acaba, entra la següent de la cua. En el cas il·lustrat, quan acabi la tasca 3 entrarà la 4, però no és possible saber què passarà després, ja que dependrà de la longitud de la nova tasca.

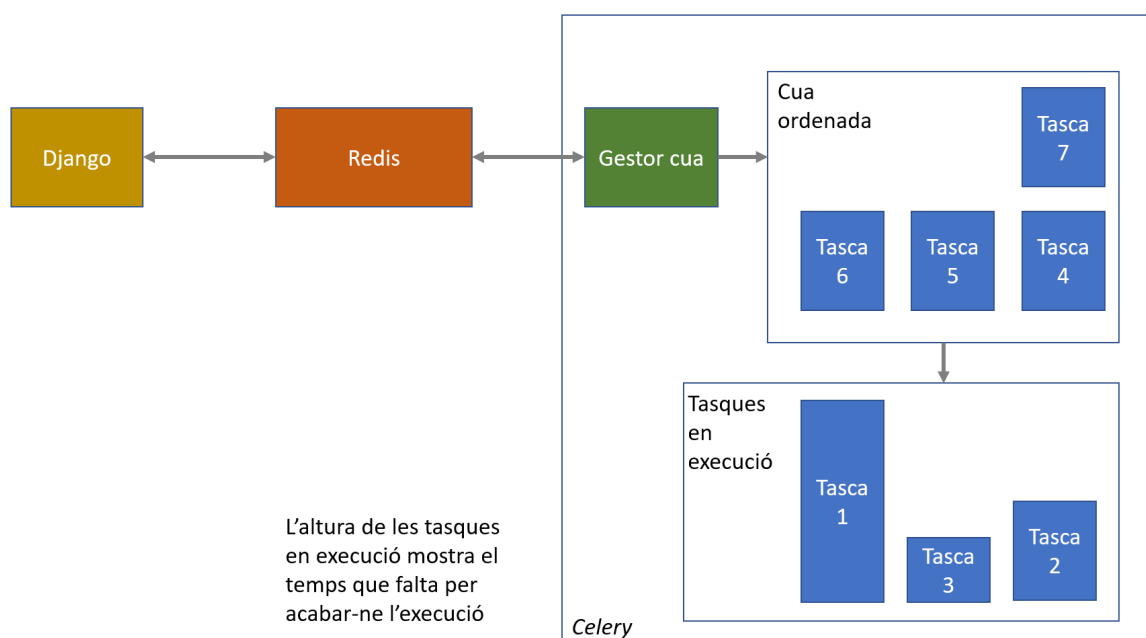


Figura 8.6: funcionament de la cua al Celery. Font: pròpia

8.2. Aplicacions

Tal i com s'ha comentat a l'apartat sobre el *Django*, s'ha creat un total de dues aplicacions que es presenten en aquest apartat.

El projecte de *Django* s'anomena *lightning*. Per poder fer referència al codi, s'utilitza el format *lightning.aplicació.fitxer.funció* i es pot trobar annexat a aquesta memòria. L'Annex C explica l'estructura dels fitxers del projecte.

8.2.1. API

Aquesta és l'aplicació amb més contingut, ja que és l'encarregada de controlar tot el dispositiu i serveix d'interfície per modificar els continguts de la base de dades a través de peticions *HTTP*.

8.2.1.1. Models

En aquest primer apartat hi ha la descripció de tots els models creats per guardar les dades necessàries pel funcionament, tant del nom que tenen com els camps creats.

- **PluggingHistory:** permet guardar totes les accions de connexió/desconnexió que s'han intentat fer i, en cas d'error, guarda el motiu.
 - **created:** (DateTimeField) hora de l'acció
 - **error:** (BooleanField) cert si l'acció no s'ha acabat satisfactòriament
 - **action:** (CharField) tipus d'acció prevista (connexió o desconnexió)
 - **message:** (TextField) missatge. Per exemple, el motiu pel qual hi ha error
- **Status:** guarda el darrer estat conegut del dispositiu. Permet saber si està connectat/desconnectat i si està previst reconnectar-lo.
 - **date:** (DateTimeField) hora en la qual el sistema es trobava en aquest estat
 - **status:** (CharField) estat del sistema (connectat o desconnectat)
 - **distance:** (FloatField) distància a la que es troba la part mòbil del SRF005
 - **reconnect_time:** (DateTimeField) hora prevista de reconexió del sistema. El camp està buit si no hi ha cap temporitzador.
 - **reconnect_task_id:** (TextField) número identificador de la tasca que reconnectarà el sistema. El camp està buit si no hi ha cap temporitzador.
- **Sky:** la predicció meteorològica del Servei Meteorològic de Catalunya (SMC) retorna un codi per informar sobre l'estat del cel. En aquesta taula hi ha l'equivalència entre el codi i l'icona i text que s'han de mostrar a la pàgina web.
 - **sky_value:** (IntegerField) codi retornat pel SMC
 - **explanation:** (TextField) text a mostrar a la pàgina web

- **storm:** (BooleanField) cert si el codi fa referència a la possible presència de llamps
- **category:** (TextField) categoria retornada pel SMC
- **smc_icon:** (URLField) URL on es troba la icona proposada pel SMC
- **icon:** (TextField) camí on es troba la icona mostrada a la pàgina web
- **Forecast:** la predicció descarregada del SMC.
 - **date:** (DateField) dia del qual és la previsió
 - **sky:** (ForeignKey) referència a un element del model *Sky* a través del valor de *sky_value*
 - **temp_max:** (FloatField) temperatura màxima prevista en graus centígrads
 - **temp_min:** (FloatField) temperatura mínima prevista en graus centígrads
 - **rain:** (FloatField) probabilitat de pluja en %
- **LightningStrike:** permet guardar els llamps detectats, tant per l'AS3935 com pel SMC.
 - **date:** (DateTimeField) hora del llamp detectat
 - **source:** (CharField) guarda si el llamp ha sigut detectat per l'AS3935 o pel SMC
 - **distance:** (FloatField) distància entre el punt de descàrrega del llamp i la masia
 - **max_current:** (FloatField) pic de corrent del llamp (només si la font és el SMC)
 - **cloud_earth:** (BooleanField) cert si la descàrrega és núvol-terra (només si la font és el SMC)
 - **latitude:** (FloatField) latitud del punt de descàrrega del llamp (només si la font és el SMC)
 - **longitude:** (FloatField) longitud del punt de descàrrega del llamp (només si la font és el SMC)

Tots aquests models es troben a *lightning.api.models* i tenen associat un serialitzador a *lightning.api.serializers*.

La majoria de models són simples i només s'informa dels camps que han de crear. En canvi, pel darrer s'ha hagut de programar que abans de guardar un nou llamp a la base de dades, primer es comprovi que existeixi una data: el SMC informa de la data, però per l'AS3935 s'ha d'introduir l'instant actual.

8.2.1.2. Rutes

Per tal de no donar un excés d'informació que pugui fer perdre la visió general, en aquest apartat únicament s'expliquen les dues rutes principals que s'han creat i que permeten desconnectar i connectar el sistema elèctric a partir d'una petició *GET* o *POST*. La resta es troben a l'annex A.

La forma d'expressar la ruta és a partir del camí. És a dir, davant de la ruta escrita cal escriure el domini (o IP) de la Raspberry Pi.

Totes les rutes mencionades es troben a *lightning.api.urls* i les vistes que generen el contingut a *lightning.api.views*.

I. Ruta /api/disconnect

Ruta principal que serveix d'índex de les rutes que permeten desconnectar el sistema elèctric. Abans de realitzar l'acció comprova que sigui necessari i executa tasques complementàries.

Accions disponibles: *GET*

a. Ruta /api/disconnect/manual

Està preparat perquè l'usuari pugui desconnectar el sistema elèctric des de la pàgina web de forma manual.

Comprova que el sistema elèctric estigui connectat i, en cas afirmatiu, fa una petició a */api/motor/unplug*.

Si es fa una petició *POST*, a més, crea un temporitzador per tal de reconnectar el sistema de forma automàtica (8.3.7) al cap de *reconnect_in* segons. En cas que aquest temporitzador ja existeixi, l'elimina i en crea un de nou amb el temps especificat.

Accions disponibles: *GET*, *POST*

Retorna els següents codis (quan es fa referència al temporitzador de reconexió només s'aplica en cas de petició *POST*):

- 201: el sistema estava desconnectat però s'ha actualitzat el temporitzador de reconexió
- 202: s'ha programat la desconexió del sistema i s'ha creat el temporitzador de reconexió.
- 409: el sistema està desconnectat. Res a fer.
- 500: hi ha hagut algun problema al crear el temporitzador de reconexió.
- 503: hi ha hagut algun problema a l'intentar programar la desconexió del sistema.

El *JSON* retornat mostra el número d'identificació de la tasca asíncrona.

b. Ruta /api/disconnect/automatic

Fa una petició *POST* a */api/disconnect/manual* amb el paràmetre *reconnect_in* que toqui. Si

s'ha fet una petició *POST*, *reconnect_in* té el valor especificat per l'usuari. En canvi, si s'ha fet una petició *GET*, té el valor per defecte d'una hora ja que és el temps que sol durar una tempesta elèctrica [56].

Està preparat per quan el dispositiu funciona de forma autònoma: envia el correu per tal que els administradors estiguin al cas que s'ha ordenat la desconnexió del sistema i si hi ha hagut algun error.

Accions disponibles: *GET*, *POST*

II. Ruta */api/connect*

És una ruta paral·lela a */api/connect*: és l'índex de les rutes que serveixen per connectar el sistema elèctric.

Accions disponibles: *GET*

a. Ruta */api/connect/manual*

Està preparat perquè l'usuari pugui connectar el sistema elèctric des de la pàgina web de forma manual.

Comprova que el sistema estigui desconnectat i, en cas que sigui així, fa una petició a */api/motor/plug*.

Si es fa una petició *POST*, a més, es pot ordenar a l'*API* que elimini el temporitzador de reconexió del sistema (8.3.7) a través del paràmetre booleà *delete_reconnection*.

Accions disponibles: *GET*, *POST*

Retorna els següents codis (quan es fa referència al temporitzador de reconexió només s'aplica en cas de petició *POST*):

- 201: el sistema estava connectat però s'ha eliminat el temporitzador de reconexió
- 202: s'ha programat la connexió del sistema i s'ha eliminat el temporitzador de reconexió.
- 409: el sistema està connectat. Res a fer.
- 500: hi ha hagut algun problema a l'eliminar el temporitzador de reconexió.
- 503: hi ha hagut algun problema a l'intentar programar la connexió del sistema.

El *JSON* retornat conté el número identificatiu de la tasca asíncrona que connecta el dispositiu.

b. Ruta /api/connect/automatic

Fa una petició *POST* a */api/connect/manual* amb *delete_reconnection=True* i retorna el mateix format de dades. Està preparat per quan el dispositiu funciona de forma autònoma: envia el correu per tal que els administradors estiguin al cas que s'ha ordenat la connexió del sistema.

Accions disponibles: *GET*

8.2.2. Pàgina web

La pàgina web és l'aplicació que l'usuari final farà servir de forma més habitual. S'ha creat una plantilla amb l'estil disponible a *MaterializeCSS* [31] seguint les indicacions de Material Design [24] de Google. La funcionalitat s'executa amb *JavaScript* i la llibreria *jQuery* permet fer peticions a l'*API* descrita a l'apartat anterior.

S'ha intentat que la pàgina s'adaptés a tots els dispositius, per fer més fàcil la navegació des de telèfons mòbils tàctils.

Disposa de quatre parts diferenciades:

1. Capçalera. En aquest cas és simple, ja que només hi ha el nom del mecanisme.



Figura 8.7: capçalera de la pàgina web. Font: pròpia

2. Control del mecanisme. Permet veure l'estat actual, connectar o desconectar el dispositiu i dades sobre la previsió meteorològica. Les diferents targetes es poden consultar en l'apartat següent.

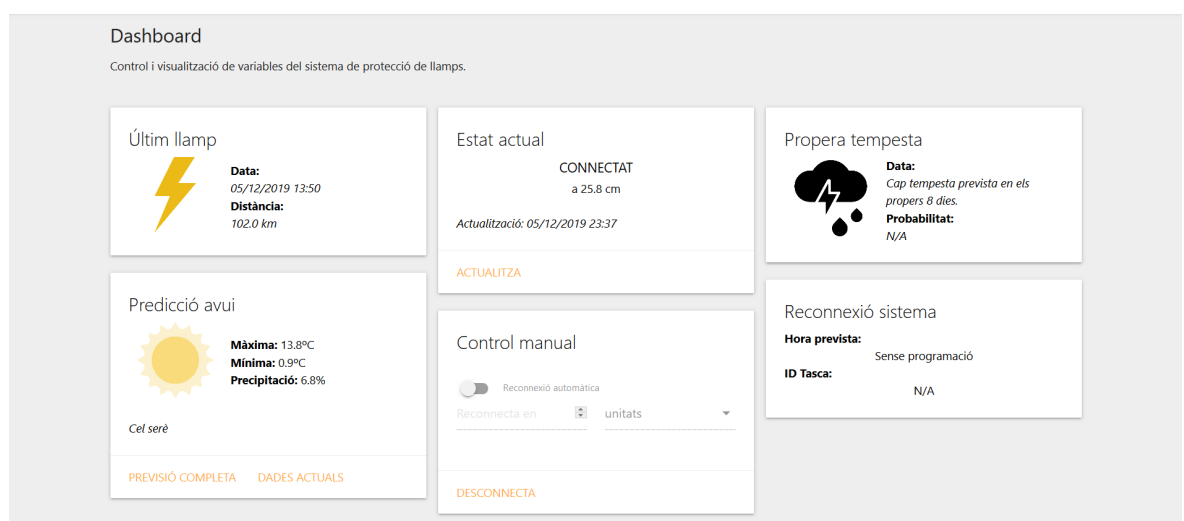


Figura 8.8: control del mecanisme des de la pàgina web. Font: pròpia

3. Dades històriques. Permet veure les darreres deu accions del model *PluggingHistory* i els deu darrers llamps detectats.

Dades històriques			
Visualització dels deu darrers valors de la base de dades			
Connexions del sistema elèctric			
Data	Acció	Error	Missatge
04/12/2019 07:16	Connectar	No	El sistema s'ha connectat correctament.
01/12/2019 21:18	Connectar	No	El sistema s'ha connectat correctament.

Llamps		
Data	Distància (km)	Font
05/12/2019 13:50	102.0	Meteo.cat
05/12/2019 13:34	67.2	Meteo.cat
05/12/2019 13:34	92.6	Meteo.cat

Figura 8.9: dades històriques mostrades a la pàgina web. Font: pròpia

4. Peu de pàgina. Mostra informació sobre el dispositiu i l'origen de les dades.

Protecció de llamps
 Control del dispositiu creat en el TFG *Mecanisme per aïllar una instal·lació durant una tempesta elèctrica.*

Universitat:

 Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

Dades extretes de:
 Servei Meteorològic de Catalunya

Desenvolupat per Eloi Codina Torras


Figura 8.10: peu de la pàgina web. Font: pròpia

Aquesta aplicació només disposa d'una ruta, */dashboard*, que permet veure la pàgina web mostrada. La vista processa la plantilla i l'omple de les dades necessàries.

La ruta es troba a *lightning.dashboard.urls* i la vista a *lightning.dashboard.views*.

8.2.2.1. Targetes disponibles

I. Últim llamp

Últim llamp


Data:
 05/12/2019 13:50
Distància:
 102.0 km

Figura 8.11: targeta "Últim llamp" de la pàgina web. Font: pròpia

Aquesta targeta mostra l'últim llamp detectat, amb la data i la distància a la que s'ha produït.

II. Predicció avui



Figura 8.12: targeta "predicció avui" de la pàgina web. Font: pròpia

Mostra la predicció meteorològica d'avui segons el Servei Meteorològic de Catalunya. L'enllaç *Previsió completa* dirigeix a la pàgina del SMC mentre que la *Dades actuals* porta a una pàgina web on es mostren les dades meteorològiques d'una estació propera a l'antena.

III. Propera tempesta



Figura 8.13: targeta "Propera tempesta" de la pàgina web. Font: pròpia

Mostra quan està previst que hi hagi una tempesta en els següents 8 dies, segons la previsió del SMC. Si no hi ha cap tempesta prevista, mostra el text *Cap tempesta prevista*.

IV. Reconnexió del sistema

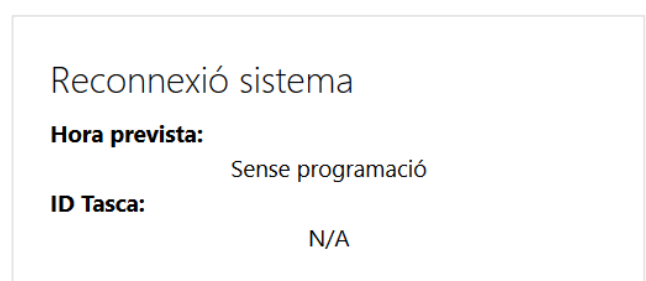
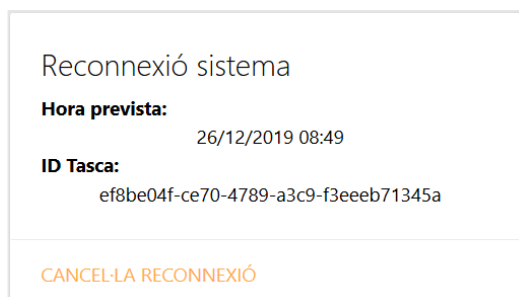


Figura 8.14: opcions de la targeta "Reconnexió sistema" de la pàgina web. Font: pròpia

Mostra si existeix un temporitzador per reconnectar el sistema elèctric. Si és així, permet

cancel·lar-lo fent una petició a `/api/status/canceltimer`.

V. Estat actual

Mostra l'estat actual del sistema i permet actualitzar-lo fent una petició a `/api/status/refresh`. En cas que no hi hagi cap estat a la base de dades, mostra el text *Si us plau, actualitza l'estat manualment*.



Figura 8.15: targeta "Estat actual" de la pàgina web.
Font: pròpia

VI. Control manual

Permet connectar i desconnectar el sistema elèctric. En cas que aquest estigui connectat, mostra les següents opcions:

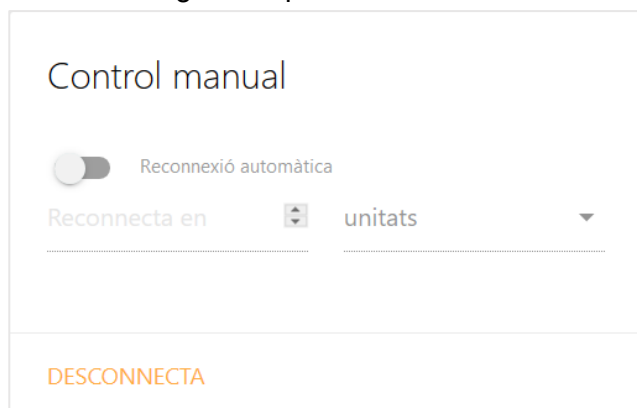


Figura 8.16: targeta "Control manual" quan el sistema es troba connectat.
Font: pròpia

Realitza una petició a `/api/disconnect/manual`. En el cas que s'especifiqui el camp *Reconnecta en*, es crearà un temporitzador per reconnectar el sistema elèctric.

En canvi, si el sistema es troba desconnectat, realitza una petició a `/api/connect/manual`. Si s'activa *Elimina el timer*, elimina el temporitzador per reconnectar el sistema elèctric en cas que existeixi.

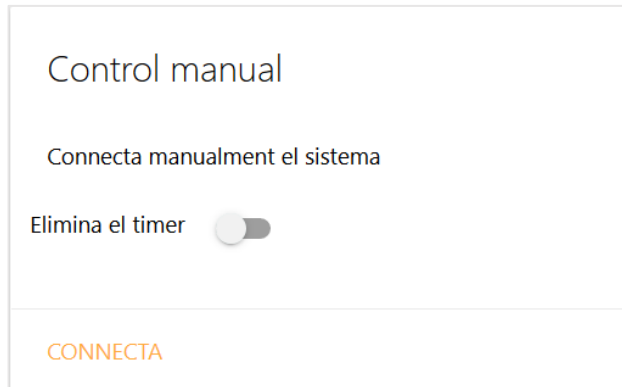


Figura 8.17: targeta "Control manual" quan el sistema es troba desconnectat. Font: pròpia

8.2.2.2. Creació d'una plantilla *HTML*

Django incorpora el mòdul *django.template* que permet llegir un fitxer *HTML* amb unes etiquetes concretes i modificar el contingut afegint el text necessari. La plantilla es troba a *lightning/dashboard/templates/index.html*.

La vista *lightning.dashboard.views.index* crea un diccionari amb totes les dades que s'han d'incorporar a la plantilla. Per exemple, si el diccionari té la següent forma:

```
{ 'status': {
    'status': 'Connectat',
    'distance': 31.8,
    'date': '20/10/2019 10:12 UTC'
  }
}
```

Llavors, si a la plantilla es posa l'etiqueta "{% status.status %}", la versió final mostrarà "Connectat".

A part, es poden posar blocs de lògica, de manera que es pot mostrar o amagar contingut segons variables. Per exemple, si hi ha el següent bloc:

```
{% if status.status == 'Connectat' %}
    Opció 1
{% else %}
    Opció 2
{% endif %}
```

La versió retornada a l'usuari mostrarà "Opció 1" si el sistema està connectat i "Opció 2" si no ho està.

8.2.2.3. Control de la pàgina

Com que les accions que es poden fer amb el mecanisme triguen un temps llarg, un cop s'ha fet la petició no es modifica el text de la pàgina sinó que es mostra un quadre indicant a l'usuari que s'ha processat la petició (i si la resposta és satisfactòria o errònia) i que la pàgina web es refrescarà automàticament al cap d'uns segons.

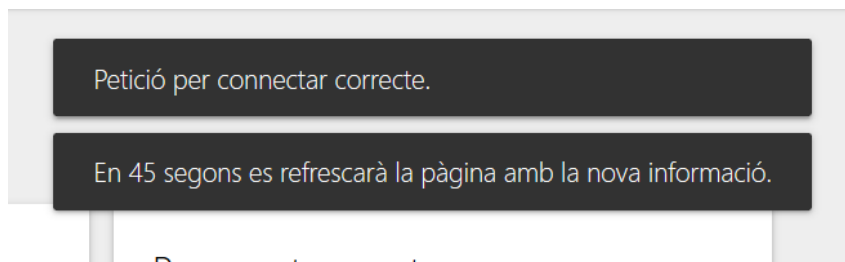


Figura 8.18: exemple de requadre mostrat per la pàgina web amb informació sobre la petició. Font: pròpia

D'aquesta manera, el contingut de la pàgina sempre és generat a través del *Django*, segons el procediment explicat en l'apartat anterior.

8.3. Tasques en segon pla

Com ja s'ha explicat anteriorment, aquelles tasques que triguen força estona a executar-se s'han creat de forma asíncrona a través de *Celery*. Totes es poden consultar a *lightning.api.tasks* i en aquest apartat es menciona què fa cada una.

8.3.1. motor_background

Aquesta tasca executa el moviment del motor a través del mòdul creat pel controlador L293D i monitoritza el sensor SRF005 per saber quan s'ha acabat el moviment. A més, per seguretat, hi ha un temporitzador per tal que si el sistema no ha acabat el procés en 45 segons, el motor s'atura automàticament.

8.3.2. send_email

Tal i com el seu nom indica permet enviar missatges de correu electrònic a través del mòdul *django.core.mail*.

8.3.3. force_status_update

Activa el sensor SRF005 i comprova la posició de la part mòbil del mecanisme per decidir si està connectat o desconnectat. A més, fa una petició a */api/status* per actualitzar l'estat.

8.3.4. **download_weather_forecast**

Aquesta funció s'executa cada dia a les 09:00 UTC i es connecta a la pàgina del Servei Meteorològic de Catalunya per descarregar la predicció dels següents vuit dies. A continuació la carrega a la base de dades a través d'una petició *POST* a */api/forecast/bulk*.

En cas que es produeixi un error, envia un correu electrònic als administradors per avisar-los que cal revisar el funcionament.

8.3.5. **check_online_lightning**

Tal i com s'ha explicat a l'apartat del sensor AS3935, aquest va arribar defectuós i no s'ha pogut provar una vegada arreglat per falta de tempestes elèctriques. Per tant, per tal de provar que el dispositiu funciona, s'ha creat aquesta funció que cada hora descarrega de l'*API* del SMC tots els llamps que hi ha hagut durant la darrera hora.

No és un funcionament òptim, ja que degut a les restriccions de l'*API*, es podria donar el cas que el sistema elèctric es desconnectés després d'una hora d'haver-se produït la primera descàrrega.

En cas que hi hagi un llamp proper, fa una petició a */api/disconnect/automatic* i envia un correu als administradors si s'ha produït un error inesperat durant la connexió amb l'*API* del SMC.

8.3.6. **as3935_monitor**

Aquesta tasca s'executa de forma contínua des del moment que el servidor Apache inicia el codi de l'aplicació. Monitoritza el sensor AS3935 i, en cas de detectar un llamp, l'afegeix a la base de dades fent una petició a */api/lightning-strikes*.

S'ha detectat que a vegades el sensor es queda penjat. Per tant, tant la tasca com el sensor es reinicien de forma automàtica cada dia a la mitjanit.

8.3.7. **reconnect_motor_timer**

Aquesta funció únicament fa una petició *GET* a */api/connect/automatic* per tal de reconnectar el sistema elèctric.

Es tracta del temporitzador de reconnexió: quan el sistema es desconnecta de forma automàtica (o manual indicant un temps de reconnexió), s'afegeix a la cua del *Celery* aquesta tasca i s'indica que cal executar-la al cap d'un temps determinat (com s'ha vist, per defecte al cap d'una hora).

8.4. Funcionament del sistema

Al capítol 5 s'ha explicat a alt nivell el funcionament del prototip. A partir del que s'ha exposat en aquest capítol, aquest apartat és més específic i mostra clarament quines són les tasques asíncrones que intervenen en el control del mecanisme.

Cal tenir present que el moviment de la part mòbil únicament és possible en els següents casos:

- El sistema està connectat
 - L'usuari desconnecta el sistema de forma manual a través de la pàgina web. Pot decidir crear un temporitzador de reconexió del sistema al cap de l'estona que desitgi.
 - Durant una de les descàrregues horàries de llamps del SMC es detecta que en la darrera hora hi ha hagut un llamp proper i, per tant, cal desconnectar el sistema. Es crea un temporitzador de reconexió del sistema (8.3.7) al cap d'una hora des de la detecció del llamp, ja que és el que sol durar una tempesta [56].
- El sistema està desconnectat
 - L'usuari connecta el sistema de forma manual a través de la pàgina web.
 - Un temporitzador per la reconexió del sistema (8.3.7), creat durant la desconnexió o actualitzat posteriorment en cas d'haver detectat algun altre llamp, ha arribat al seu final i, per tant, cal reconnectar el sistema elèctric ja que se suposa que la tempesta ja ha passat.

A més, les següents tasques principals s'executen de forma contínua:

- Descàrrega horària de llamps: tal i com s'ha vist a l'apartat 8.3.5, un cop cada hora es descarreguen els llamps de l'hora anterior de l'*API* del Servei Meteorològic de Catalunya. Si procedeix, s'inicia la desconnexió o s'actualitza el temporitzador de reconexió del sistema (8.3.7).
- Monitor de l'AS3935: vist a l'apartat 8.3.6, en cas que l'AS3935 detecti un llamp, aquesta és la tasca asíncrona que afegirà les dades a la base de dades.
- Descàrrega diària de la predicció meteorològica: cada dia a les 09:00 UTC es descarrega la predicció meteorològica dels següents vuit dies de l'*API* del SMC. És la funció vista a l'apartat 8.3.4.

9. Gestió del projecte

En aquest projecte s'ha posat especial èmfasi en la gestió de projectes informàtics amb la filosofia *Agile* [3] i la metodologia *Scrum* [48] que, malgrat estar dissenyades per funcionar en equips, s'han adaptat per treballar de forma individual.

En aquest capítol s'explica com s'ha treballat tant en la part informàtica com en el projecte global, així com les eines que s'han usat per desenvolupar el programa.

9.1. Programes emprats durant el desenvolupament

9.1.1. PyCharm

Per tal de programar en *Python* s'ha fet servir l'entorn de desenvolupament integrat de JetBrains, *PyCharm* [28], amb la seva llicència acadèmica. Es tracta d'un editor de text que té validació de codi i ajuda a la programació ja que té la funcionalitat d'autocompletar les línies i proposar funcions de mòduls importats.

9.1.2. Connexió SSH: MobaXterm

Al treballar des d'un entorn Windows ha sigut necessari buscar un terminal avançat amb connexió *SSH*. *MobaXterm* [33] és un dels terminals més ben valorats per accedir a ordinadors remots sense interfície gràfica.

A més, disposa d'una interfície gràfica d'*SFTP* que permet interactuar amb els fitxers que es troben en els directoris de la Raspberry Pi amb el ratolí i copiar-los a l'escriptori.

9.1.3. Control de versió: Git

Qualsevol projecte d'informàtica fa servir algun tipus de control de versió per tal de tenir traçabilitat dels canvis realitzats i una còpia de seguretat del codi. En el cas d'aquest treball s'ha fet servir el servidor de *GitLab* [22] que permet tenir repositoris (projectes) privats de forma gratuïta.

Cada cop que s'acabava la programació d'una funcionalitat, a través de la línia de comandos de la Raspberry Pi, s'enviava el codi al servidor. En projectes grans s'usen branques [10] diferents per poder programar de forma simultània sense que el canvi que fa un programador trenqui els canvis d'un altre o provar funcionalitats noves. Per aquest projecte, al treballar de forma individual, només s'ha fet ús de la branca principal, *master*.

9.2. Gestió global del projecte

El següent diagrama de Gantt mostra com s'han repartit les hores per dur a terme el projecte:

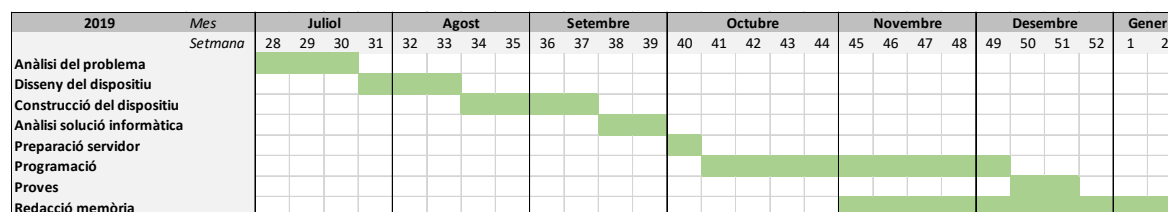


Figura 9.1: diagrama de Gantt. Font: pròpia

Per fer el pressupost és necessari conèixer les hores invertides. La següent taula resumeix, per a cada activitat, el número d'hores que s'han necessitat:

Activitat	Hores
Anàlisi del problema	20
Disseny del dispositiu	25
Construcció del dispositiu	50
Anàlisi solució informàtica	25
Preparació servidor	8
Programació	72
Proves	5
Redacció memòria	80
TOTAL	284

Taula 9.1: hores dedicades a cada activitat del projecte. Font: pròpia

Com es pot observar, l'inici del treball va ser just després dels exàmens finals del quadrimestre de primavera del curs 2018-2019. El primer pas va ser analitzar el problema durant tres setmanes, que en aquesta memòria ha quedat reflectit en el primer apartat.

A continuació es va dissenyar i construir el prototip, tal i com s'ha explicat a l'apartat 6. El temps total va ser d'unes set setmanes.

La següent part és la que en proporció ha portat més feina: com controlar el prototip construït. Analitzar com els diferents sensors i controladors havien d'interactuar va necessitar de dues setmanes i preparar la Raspberry Pi, així com totes les eines emprades, una setmana.

Per la part informàtica s'han invertit un total de 80 hores repartides en nou setmanes, temps que inclou també la documentació de tot el codi. Es coneix el valor amb força precisió ja que tal i com es veurà en el següent apartat, s'han usat eines per controlar el temps dedicat.

9.3. Gestió de la part informàtica: *Agile* i *Scrum*

Durant el desenvolupament del treball s'ha estat treballant paral·lelament en una empresa on se segueix la filosofia *Agile*. Per tal d'aprofundir els coneixements, s'ha decidit intentar adaptar-la per crear la part informàtica.

Scrum és la definició de metodologies basades en *Agile* per a dur a terme el desenvolupament de programari. Existeixen diferents paradigmes, però s'ha escollit aquest ja que és l'usat en l'empresa.

Agile es basa en quatre valors publicats a l'*Agile Manifesto*:

- Interaccions entre individus enlloc de processos i eines
- Programari funcional en lloc de documentació exhaustiva
- Col·laboració del client en lloc de negociacions de contracte
- Adaptar-se als canvis en lloc de seguir un pla

Per aquest projecte el darrer valor ha sigut d'alta importància, doncs el producte informàtic final ha anat variant segons avançava el desenvolupament. Per tal d'aconseguir aquesta adaptació s'ha usat el concepte d'iteració: en un temps determinat calia finalitzar noves funcionalitats per tal de tenir un model de desenvolupament continu. Al final de cada iteració es dedicava un temps a reflexionar sobre què s'havia aconseguit i cap a on anava el projecte.

9.3.1. Eines utilitzades

9.3.1.1. YouTrack

YouTrack [29] és un programari de JetBrains que permet gestionar les tasques que s'han de realitzar en un entorn *Agile*. Té una versió en línia gratuïta per a fins a tres usuaris i s'ha utilitzat per poder classificar i saber totes les funcionalitats que calia desenvolupar per arribar al final del projecte. A més, a cada tasca s'ha afegit el temps emprat, de manera que es pot conèixer el temps dedicat a cada part del programari desenvolupat.

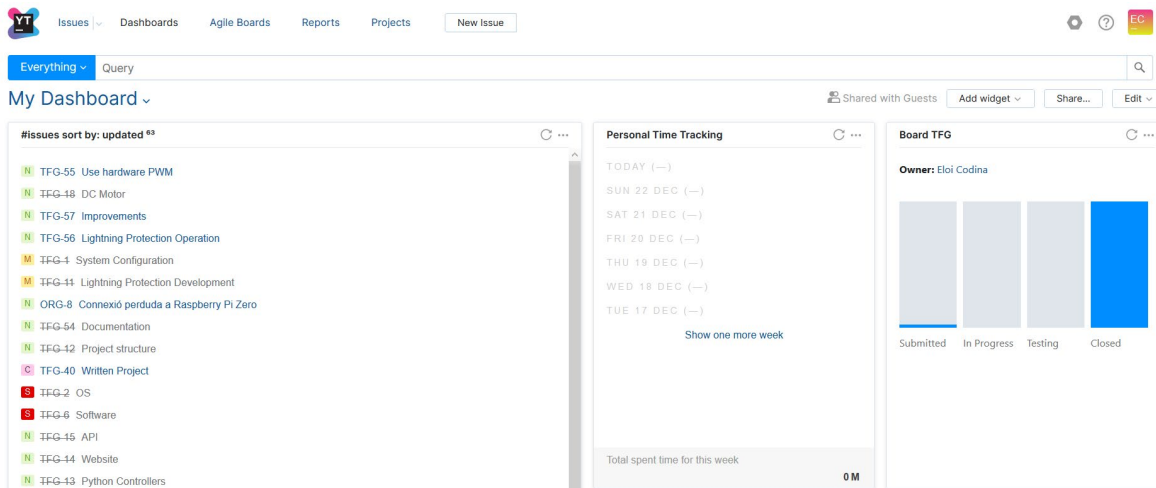


Figura 9.2: captura de pantalla de YouTrack. Font: pròpia

9.3.1.2. KanBan

El KanBan no és res més que un panell on apareixen les tasques que s'estan realitzant, les que ja s'han acabat i les que estan per començar. *YouTrack* disposa d'un KanBan virtual, però en aquest projecte s'ha preferit tenir-lo en format físic i afegir *post-its* per a cada tasca. Amb aquesta eina es planificaven les funcionalitats que es desenvoluparien en cada iteració.

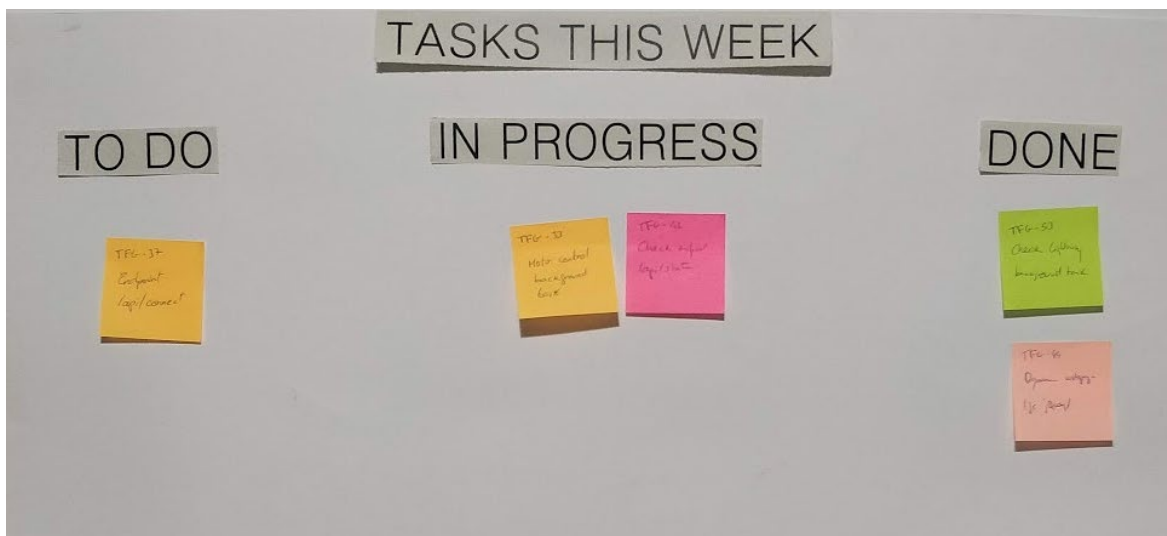


Figura 9.3: KanBan utilitzat durant el desenvolupament. Font: pròpia

9.3.2. Estructura de les tasques

Per tal de dividir la feina i determinar l'abast del projecte s'ha començat enumerant els objectius a assolir des d'una perspectiva gran i s'han anat dividint fins a aconseguir tasques petites que es podien desenvolupar en una iteració i aporten valor.

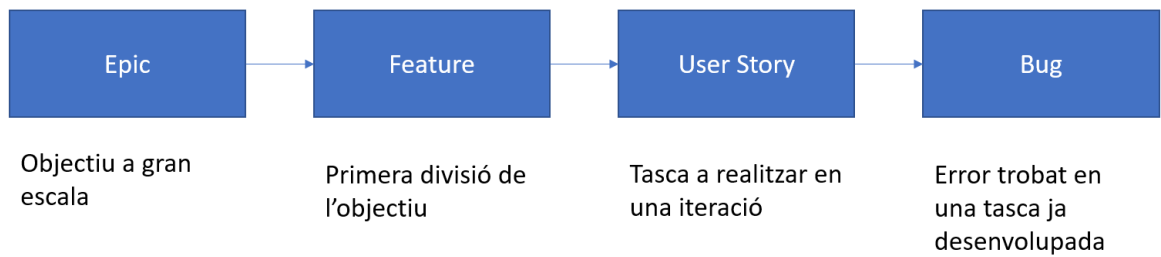


Figura 9.4: estructura de les tasques. Font: pròpia

Aquestes estructures s'han generat des d'un punt de vista empresarial, és a dir, buscar què vol l'usuari i, a partir d'aquestes necessitats, crear tasques tècniques dependents. Per exemple, la pàgina web seria una *Epic*, cada part (capçalera, zona de control, etc), seria una *Feature* i cada component seria una *User Story*. Aquestes *User Story* en generarien d'altres tècniques que serien les diferents rutes de l'*API*, ja que són necessàries per tal d'aconseguir la funcionalitat desitjada.

De cada tasca s'ha valorat l'esforç de forma qualitativa indicant una talla de samarreta. D'aquesta manera, abans de començar una iteració es podia saber si es disposaria de temps suficient per acabar el desenvolupament o no.

A part, totes les tasques estaven prioritzades per tal de començar amb aquelles més rellevants: no aporta el mateix valor que al peu de la pàgina web aparegui el logotip del Servei Meteorològic de Catalunya que hi hagi una targeta que permeti desconnectar el sistema elèctric.

Una *User Story* ha de ser independent, valuosa, estimable, petita i que es pugui provar. A més, ha d'incorporar una descripció que expliqui què es vol aconseguir i el perquè, des del punt de vista de l'usuari. També cal explicitar els criteris d'acceptació, és a dir, com saber que la tasca desenvolupada funciona correctament i es pot donar per acabada.

Tornant a l'exemple de la pàgina web, la definició de la història d'usuari referent a la targeta que permet desconnectar el sistema elèctric, seria: *com a usuari vull una targeta que permeti desconnectar el sistema elèctric, definint el temps del temporitzador de reconexió*. A continuació s'exposarien els criteris d'acceptació, com ara: *cal que la pàgina mostri el resultat de l'operació i, en cas d'error, mostri el missatge X*.

La següent captura de pantalla del *YouTrack* mostra la definició de la *User Story* tècnica dependent d'aquesta, és a dir, la ruta de l'*API* que permet desconnectar el sistema. Es pot observar que conté els mateixos elements que la descrita anteriorment.

Endpoint to disconnect electric system



A call to this endpoint should either start the DC motor to disconnect the system (via the `/api/motor` endpoint), restart the reconnect timer or return an error to the website.

It comprises all the logic behind disconnecting the electric system.

Endpoint: `/api/disconnect`

[GET]

- `/api/disconnect/manual/`: to disconnect without any reconnection timer and without sending an email.
- `/api/disconnect/automatic/`: to disconnect with a reconnection timer with the default time and send a warning email if the system was connected.

[POST]

- `/api/disconnect/manual/`: to disconnect with a reconnect timer in the seconds specified by the user. Without email warning
 - `reconnect_in`: seconds until reconnection.
- `/api/disconnect/automatic/`: to disconnect with a reconnect timer in the seconds specified by the parameter. With email warning
 - `reconnect_in`: seconds until reconnection.

Possible returns for `/api/disconnect/manual`

[GET]

- 202. ■ `{'status': 'Disconnecting', 'motor_task_id': 'asdf-234-qwef-'}`
- 409. ■ `{'status': 'Already disconnected'}`
- 503. ■ `{'status': 'Error disconnecting', 'motor_error': 'It failed because...'}`

Figura 9.5: exemple de User Story. Font: pròpia

Totes les tasques formen part del *Product Backlog* que, tal i com el seu nom indica, conté tota la informació necessària per a dur a terme el projecte. A l'inici de cada iteració, tal i com es veurà en un apartat posterior, es creava l'*Iteration Backlog*, que és el llistat de tasques a desenvolupar durant l'iteració segons la prioritització del *Product Backlog*.

9.3.3. Flux de treball

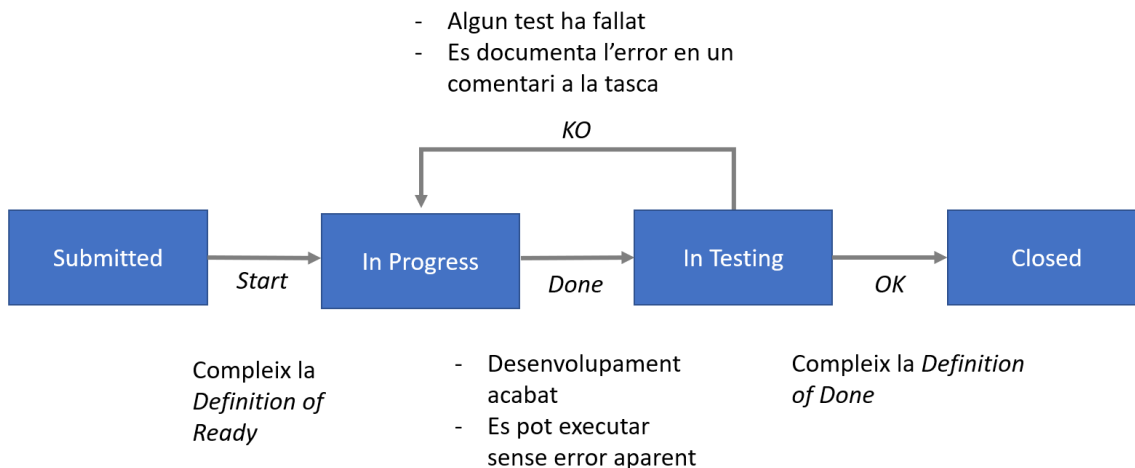


Figura 9.6: flux de treball en Agile. Font: pròpia

A la figura anterior es pot veure quins són els estats pels quals passa qualsevol tasca. Hi ha dos artefactes importants, que són les següents definicions:

1. **Definition of Ready:** defineix quan una tasca pot ser considerada per ser desenvolupada en una iteració. En aquest projecte s'ha definit com:
 - a. *Tasca clarament definida, amb els criteris d'acceptació visibles*
 - b. *És factible acabar el desenvolupament en la següent iteració*
2. **Definition of Done:** en aquest cas la definició explícita quan una tasca pot ser considerada acabada:
 - a. *Desenvolupament acabat*
 - b. *Codi documentat*
 - c. *Tests basats en els criteris d'acceptació es completen correctament*
 - d. *S'ha enviat el codi al servidor de GitLab*

A part, hi ha un estat que no es mostra a la figura: pot ser que una tasca es cancel·li perquè ja no aporta valor. En el cas d'aquest treball, es volia poder usar el mecanisme a través de comandes per veu a través de Google Home, però finalment es va cancel·lar ja que no tenia molt sentit.

9.3.4. Iteracions i litúrgies

En aquest darrer apartat sobre *Agile* es presenta com s'han dissenyat les iteracions. Tot i que la guia d'*Scrum* recomana fer iteracions d'entre dues i quatre setmanes, s'ha decidit que fossin d'una setmana.

Totes començaven un dilluns i acabaven el diumenge i entremig es feien dues litúrgies importants:

1. **Iteration Planning:** es tracta de preparar l'*Iteration Backlog*, és a dir, tota la feina que es farà en aquella iteració. A partir de la valoració de cada tasca se n'agafaven tantes com es poguessin abordar (normalment unes tres) començant per aquella que tingui la prioritat més elevada. Les escollides s'escriuen en un post-it i s'enganxaven a la part *TO DO* del *KanBan*.
2. **Iteration Review:** en aquesta litúrgia s'han ajuntat dues de les que proposa *Scrum*, *Review* i *Retrospective*. Es busca fer una reflexió sobre com ha anat la iteració, quines millores es poden fer i donar una finalització oficial. Tasques a realitzar:
 - a. Identificar les tasques no finalitzades: aquestes passen directament a la següent iteració
 - b. Identificar noves tasques que no es troben en el *Product Backlog* i que cal afegir
 - c. Decidir si hi ha alguna tasca del *Product Backlog* que ja no és necessària i es pot cancel·lar
 - d. Treure tots els post-its del *KanBan*
 - e. Comprovar que tot el codi desenvolupat es troba a GitLab

El següent diagrama mostra com és aquest flux:

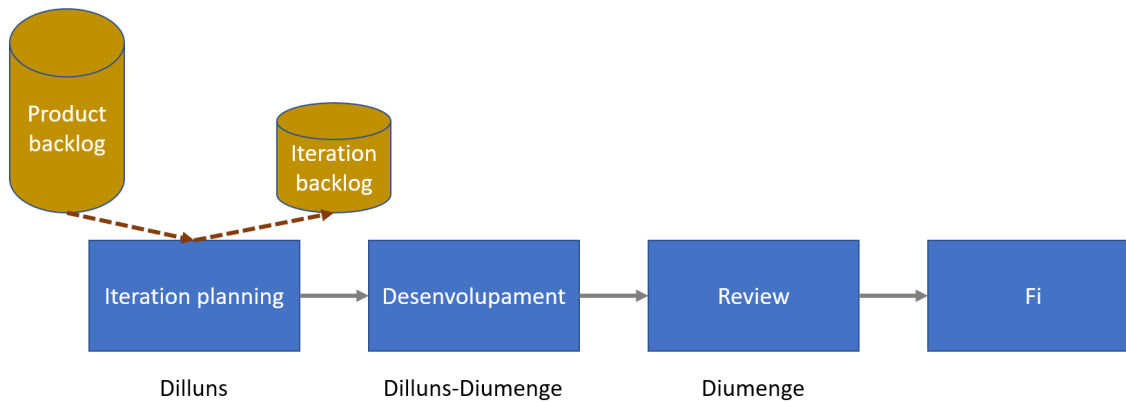


Figura 9.7: una iteració. Font: pròpia

10. Anàlisi i mitigació de riscos

Com en qualsevol producte, el prototip construït comporta una sèrie de riscos. Conèixer-los ajuda a mitigar les conseqüències derivades de l'ús d'aquest.

1. L'usuari es pot electrocutar ja que no hi ha els terminals elèctrics protegits.

Està previst que en una propera iteració del prototip, s'afegeixi una placa de metacrilat transparent a sobre del prototip, impedint l'accés a la part mòbil i als terminals elèctrics però mantenint la visibilitat del mecanisme. D'aquesta manera l'usuari no podria tocar els terminals de forma involuntària.

2. Perill d'incendi del dispositiu. Una guspira elèctrica podria encendre el prototip.

Està fabricat amb PVC, que compleix la normativa vigent segons la norma UNE 53.315 [6] i està classificat com a material difícilment inflamable. En el cas que una guspira provoqués un petit foc, aquest s'autoextingiria.

3. La font d'alimentació del motor deixa de funcionar.

Això provocaria que el mecanisme no es pogués moure. Com que el sensor de distància no detectaria cap moviment, enviaria un correu electrònic a l'usuari per tal que comprovi què ha passat de forma manual.

4. La part mòbil del dispositiu queda encallada.

Similar al risc anterior però amb la particularitat que el motor seguiria alimentat. S'ha programat un temporitzador de 45 segons: si no ha acabat el moviment en aquest temps, s'atura el motor i envia un missatge a l'usuari per tal que comprovi què ha passat de forma manual.

5. El sensor ultrasònic de distància no funciona correctament.

En aquest cas no es detectaria el moviment. Com en el risc anterior, el temporitzador aturaria el moviment al cap de 45 segons i avisaria a l'usuari a través de correu electrònic.

6. La Raspberry Pi perd l'alimentació i s'apaga bruscament

Quan un ordinador s'apaga sense avís pot provocar que els fitxers quedin malmesos. Per tant, en aquest risc hi ha dos casos diferenciats:

- a. La targeta SD queda corrompuda

Tot el codi es troba a *GitLab* i es té una còpia de seguretat de la SD. Així, només caldria copiar la còpia a la SD i tornaria a funcionar. Es perdrien les dades històriques (descàrregues elèctriques detectades, accions, etc) però la funcionalitat seguiria intacta.

b. Es pot reiniciar correctament la Raspberry Pi

La Raspberry Pi està programada per tal que quan s'engega inici el codi desenvolupat en aquest treball de forma automàtica. A més, el *Redis* (missatger per a les tasques asíncrones) té persistència. Per tant, totes aquelles tasques que es trobaven en cua o que la seva execució encara no s'havia acabat, s'executarien de forma normal. La funcionalitat no es veuria afectada i l'usuari no hi hauria d'intervenir.

11. Pressupost

En aquest capítol es presenta el pressupost del projecte desglossat per temes. Tots els preus són sense IVA.

11.1. Cost del material

Una part important ha sigut la construcció del prototip que, tot i tenir gran part dels materials reaprofitats, s'han valorat a partir de preus trobats a internet. La següent taula mostra els materials que s'han necessitat.

Concepte	Unitats	Preu/unitat	Total
Part física			100,18
Planxa PVC 500x500x18 mm	1	97,15	97,15
Vares fibra de vidre i polièster 2 m x	1	0,65	0,65
Cola PVC	1	1,50	1,50
Cartolina	1	0,10	0,10
Cargols de diferents mides	14	0,04	0,56
Femelles de diferents mides	8	0,02	0,16
Volanderes de diferents mides	6	0,01	0,06
Part elèctrica i motriu			27,95
Base 3 endolls	1	2,80	2,80
Cable elèctric (metres)	0,7	1	0,70
Terminal d'alumini	2	0,10	0,20
Terminal molles	2	0,50	1
Conjunt motor- corretja (reutilitzat d'una impressora)	1	5	5
Font alimentació motor	1	8,50	8,50

<i>Font alimentació Raspberry Pi</i>	1	9,75	9,75
Part electrònica			86,61
<i>Raspberry Pi 3</i>	1	30,02	30,02
<i>Targeta SD 32 GB</i>	1	8,50	8,50
<i>Carcassa protecció Raspberry Pi</i>	1	4,50	4,50
<i>Protoboard</i>	1	3,50	3,50
<i>Resistència 1 kΩ</i>	3	0,03	0,09
<i>Paquet 10 cables de prototipatge</i>	1	5,50	5,50
<i>Controlador L293D</i>	1	0,80	0,80
<i>Sensor llamps MOD-1016/AS3935</i>	1	27,30	27,30
<i>Sensor distància ultrasònic SRF005</i>	1	6,40	6,40
TOTAL			214,74

Taula 11.1: cost del material. Font: pròpia

11.2. Cost del personal

Tal i com s'ha vist a l'apartat 9.2, per dur a terme aquest projecte s'han necessitat 284 hores. A partir de les indicacions del Director del treball, s'ha considerat un preu de 35€/hora.

Activitat	Hores	Preu (€)
<i>Anàlisi del problema</i>	20	700
<i>Disseny del dispositiu</i>	25	875
<i>Construcció del dispositiu</i>	50	1750
<i>Anàlisi solució informàtica</i>	25	875
<i>Preparació servidor</i>	8	280

<i>Programació</i>	72	2520
<i>Proves</i>	5	175
<i>Redacció memòria</i>	80	2800
TOTAL	284	9940

Taula 11.2: cost del personal. Font: pròpia

11.3. Cost de l'equipament

Durant el procés de construcció del prototip s'han fet servir serres manuals, un trepant de sobretaula, serjants, tornavisos i llimes. S'ha fet una estimació de l'amortització d'aquestes.

A part, s'ha usat un ordinador portàtil per analitzar, dissenyar i redactar la memòria. També s'ha fet una estimació de l'amortització.

Pel que fa a les llicències de programari només apareix la versió acadèmica de Microsoft Office ja que tant el *SolidWorks*, *PyCharm*, *MobaXterm*, *GitLab*, *YouTrack* com altres programes utilitzats disposen de versió gratuïta.

<i>Equipament</i>	<i>Cost total (€)</i>	<i>Vida prevista (mesos)</i>	<i>Temps usat (mesos)</i>	<i>Cost (€)</i>
<i>Trepant</i>	280	120	1	2,35
<i>Conjunt eines</i>	50	120	1	0,42
<i>Ordinador</i>	1200	24	6	300
<i>Microsoft Office</i>	70	48	6	8,75
TOTAL				311,52

Taula 11.3: cost de l'equipament. Font: pròpia

11.4. Cost energètic

L'últim cost associat a la creació del prototip d'aquest treball és el derivat del consum energètic. S'ha considerat que el consum té un preu de 0,17 €/kWh [49], incloent el terme de

potència contractada a la llar.

Concepte	Potència (W)	Hores	Cost (€)
<i>Ordinador</i>	150	234	5,97
<i>Raspberry Pi</i>	5,32	85	0,08
<i>Trepant</i>	800	1	0,14
<i>Il·luminació (dues bombetes de 6 W)</i>	12	284	0,58
TOTAL			6,77

Taula 11.4: cost energètic. Font: pròpia

11.5. Cost total

La següent taula resumeix el cost total del projecte

Concepte	Cost total (€)
<i>Materials</i>	214,74
<i>Personal</i>	9940
<i>Equipament</i>	311,52
<i>Consum energètic</i>	6,77
TOTAL	10473,03

Taula 11.5: cost total. Font: pròpia

12. Impacte ambiental

Tal i com s'ha comentat en el capítol de la construcció del prototip, la majoria dels materials usats són reaprofitats i, per tant, amb aquest treball s'han reduït residus. Tot i això, en aquest capítol s'estudia l'impacte que podria tenir el dispositiu.

12.1. Consum energètic

El consum del prototip en funcionament correspon únicament al de la Raspberry Pi amb una potència aproximada de 5 W. Es tracta d'un consum molt baix i com que la masia funciona amb energia solar, l'impacte ambiental es pot considerar nul.

Pel que fa al disseny, construcció i proves s'ha fet servir energia provinent de la xarxa elèctrica i segons Red Eléctrica de España [43] la distribució de la producció és la següent:

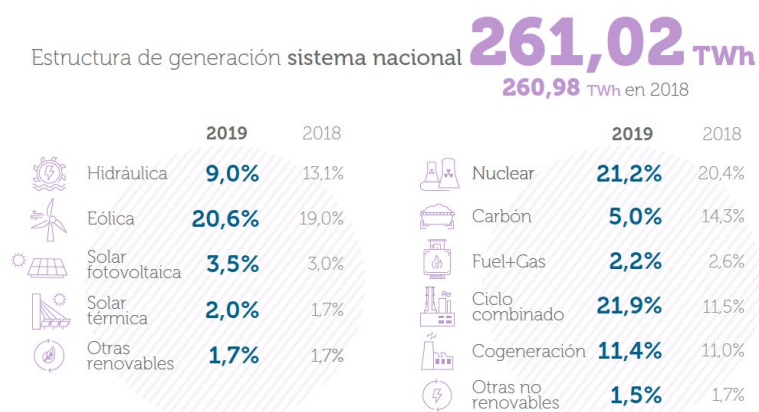


Figura 12.1: producció d'energia elèctrica a l'estat espanyol. font: ree.es

Segons dades de la Generalitat de Catalunya [37] calculades a partir de la distribució anterior, per aquest any es pot considerar un factor d'emissió de 321 g CO₂/kWh. Al pressupost s'ha calculat que s'ha consumit aproximadament 40 kWh i, per tant, l'empremta del carboni de l'energia consumida és de 12,8 kg de CO₂.

12.2. Materials

Bàsicament s'han usat dos materials: PVC i alumini. Segons un informe de la Comissió Europea [19], la producció de CO₂ per a cada un és de 0,85 kg CO₂/kg PVC i 2,57 kg CO₂/kg Al. A més, el PVC es pot reciclar fins a un 70% i l'alumini fins a un 85%.

La quantitat d'alumini usat és menyspreable. Pel que fa al PVC, com que el prototip té una massa d'uns 2 kg, per produir-lo s'han emès 1,70 kg CO₂. Durant la fase de producció també s'emeten quantitats importants de SO₂ i NO_x, gasos d'efecte hivernacle i contaminants.

Ara bé, no es pot perdre de vista que es tracta de material reaprofitat, que ja havia acabat la seva vida útil i s'ha reutilitzat en una nova aplicació. Així doncs, l'impacte ambiental és baix doncs s'ha evitat que entressin en el procés de gestió de residus.

12.3. Components electrònics

L'últim gran grup que cal considerar són els diferents components electrònics. No s'ha trobat un estudi d'impacte ambiental d'ells però es pot assegurar que compleixen la normativa vigent ja que són comercialitzats a tota la Unió Europea.

No per això es pot menysprear el seu impacte. En termes de l'empremta de carboni, segons NXP [36], durant l'any 2018 es van emetre 20 tones de CO₂ per a cada metre de làmina de silici produït. Aquestes làmines serveixen de base per crear els xips, de manera que l'emissió total per fabricar els sensors i controladors és encara més gran que aquest valor.

A més, el reciclatge de circuits integrats és complicat. Per tant, es pot dir que l'impacte ambiental degut als components electrònics és rellevant i s'hauria d'estudiar si és possible canviar algun sensor digital per un altre de mecànic que tingui un impacte menor.

Conclusions

En aquest projecte s'ha fet una anàlisi suficientment completa dels efectes d'un llamp sobre la instal·lació de la muntanya per entendre quins requisits necessitava complir el prototip. Tot i que podria ser més exhaustiva, dona una visió global sobre la qual treballar.

S'ha aconseguit construir un dispositiu funcional d'acord amb les restriccions trobades tot i que, malauradament, el sensor de tempestes no ha funcionat com estava esperat: tot i estar connectat, la falta de llamps no ha permès conèixer la seva fiabilitat. Aquest fet no ha impedit acabar el projecte ja que el funcionament autònom es basa en les dades del Servei Meteorològic de Catalunya.

També s'ha demostrat que una Raspberry Pi és un ordinador versàtil i suficientment potent per aplicacions en prototips i s'han augmentat els coneixements d'informàtica, en especial la programació d'APIs amb *Django* i la comunicació amb sensors digitals.

A més, s'ha pogut posar en pràctica *Agile* en un entorn unipersonal, aprofundint les nocions que es tenien i que poden ser útils en un entorn professional. D'aquesta manera, s'ha donat força pes a la vessant informàtica i de gestió de projectes, que és el que més es volia treballar.

Per tot això, una vegada finalitzat el projecte, es pot afirmar que s'han assolit els objectius proposats a la introducció.

De cara al futur, seria interessant fer un estudi més extens de l'anàlisi del problema per poder comprovar que el prototip proposat és realment efectiu per evitar sobretensions a la instal·lació de la masia. En el capítol corresponent s'han citat articles que poden servir de base per a tal estudi.

A part, caldria polir els detalls exposats durant l'anàlisi de riscos per aconseguir un producte final més segur i seria interessant substituir la *ProtoBoard* per un circuit imprès.

Agraïments

En primer lloc, agrair al director del treball, Lluís Solano, haver acceptat la meva proposta, així com per la guia proporcionada durant el desenvolupament del projecte.

A la meva família, per proporcionar-me una problemàtica en la que basar el Treball Fi de Grau, així com pel cop de mà durant la construcció del prototip.

Voldria agrair a l'equip d'Epic Owners d'Allianz Technology, i companys meus durant sis mesos, Darío Peña, Rebeca Bareiro i Daniel Kraus, per l'ajuda rebuda a l'hora d'adaptar *Agile* en un entorn unipersonal.

Finalment, agrair la predisposició del Servei Meteorològic de Catalunya, que ha ofert accés a la seva API amb dades horàries de descàrregues elèctriques atmosfèriques de forma gratuïta.

Bibliografia

- [1] D. A. Kohl, "A 500-kHz Sferics Range Detector," *J. Appl. Meteorol.*, p. 8, 1969.
- [2] J. Abyz, "pigpio library." [Online]. Available: <http://abyz.me.uk/rpi/pigpio/>. [Accessed: 22-Dec-2019].
- [3] Agile Alliance, "What is Agile Software Development?" [Online]. Available: <https://www.agilealliance.org/agile101/>. [Accessed: 23-Dec-2019].
- [4] ams AG, "AS3935: Franklin Lightning Sensor IC," 2016. [Online]. Available: https://ams.com/documents/20143/36005/AS3935_DS000385_1-00.pdf. [Accessed: 16-Dec-2019].
- [5] Arduino AG, "Arduino Uno Rev3 | Arduino Official Store." [Online]. Available: <https://store.arduino.cc/arduino-uno-rev3>. [Accessed: 16-Dec-2019].
- [6] Asociación Ventanas PVC, "El PVC y el fuego - Ventanas y Cerramientos." [Online]. Available: https://www.interempresas.net/Cerramientos_y_ventanas/Articulos/43213-El-PVC-y-el-fuego.html. [Accessed: 23-Dec-2019].
- [7] S. Ben Avraham, "What is REST — A Simple Explanation for Beginners, Part 2: REST Constraints," 2017. [Online]. Available: <https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-2-rest-constraints-129a4b69a582?> [Accessed: 22-Dec-2019].
- [8] Broadcom, "BCM2835 ARM Peripherals." [Online]. Available: <https://www.raspberrypi.org/app/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>. [Accessed: 16-Dec-2019].
- [9] C. Castiglione, "10 Famous Websites Built Using Python." [Online]. Available: <https://learn.onemonth.com/10-famous-websites-built-using-python/>. [Accessed: 22-Dec-2019].
- [10] S. Chacon, "Git - git-branch Documentation." [Online]. Available: <https://git-scm.com/docs/git-branch>. [Accessed: 23-Dec-2019].
- [11] D. Crockford, "Introducing JSON." [Online]. Available: <https://www.json.org/json-en.html>. [Accessed: 22-Dec-2019].
- [12] B. Croston, "RPi.GPIO · PyPI." [Online]. Available: <https://pypi.org/project/RPi.GPIO/>. [Accessed: 22-Dec-2019].
- [13] curl, "curl." [Online]. Available: <https://curl.haxx.se/>. [Accessed: 22-Dec-2019].
- [14] Debian, "Apt - Debian Wiki," 2019. [Online]. Available: <https://wiki.debian.org/Apt>. [Accessed: 16-Dec-2019].
- [15] L. Diaz, M. Martinez, J. Ramírez, and M. Rubinstein, "Lightning induced voltages in simple configurations of underground cables," in *2012 31st International Conference on*

Lightning Protection, ICLP 2012, 2012.

- [16] Django Software Foundation, “Django web framework.” [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 16-Dec-2019].
- [17] Eland Cables, “FAQ: The benefits of PVC insulated cables.” [Online]. Available: <https://www.elandcables.com/the-cable-lab/faqs/faq-what-are-the-benefits-of-pvc-insulated-cables>. [Accessed: 28-Dec-2019].
- [18] L. Encode OSS, “Django REST framework.” [Online]. Available: <https://www.django-rest-framework.org/>. [Accessed: 22-Dec-2019].
- [19] European Comission, “Life Cycle Assessment of PVC and of principal competing materials,” 2004. [Online]. Available: https://ec.europa.eu/environment/waste/studies/pdf/pvc-final_report_lca.pdf.
- [20] Ezoic Limited, “I2C Bus, Interface and Protocol.” [Online]. Available: <https://i2c.info/>. [Accessed: 21-Dec-2019].
- [21] B. Francis, “How to build your own private smart home with a Raspberry Pi and Mozilla’s Things Gateway -,” 2018. [Online]. Available: <https://hacks.mozilla.org/2018/02/how-to-build-your-own-private-smart-home-with-a-raspberry-pi-and-mozillas-things-gateway/>. [Accessed: 16-Dec-2019].
- [22] GitLab, “About GitLab.” [Online]. Available: <https://about.gitlab.com/>. [Accessed: 23-Dec-2019].
- [23] Gómez Maqueda S.A., “Conductores de cobre VS conductores de aluminio.” [Online]. Available: <https://www.gomezmaqueda.com/conductores-cobre-aluminio/>. [Accessed: 28-Dec-2019].
- [24] Google, “Material Design.” [Online]. Available: <https://material.io/design/>. [Accessed: 22-Dec-2019].
- [25] L. Grigorescu and S. Nastac, “THEORETICAL AND PRACTICAL ASPECTS OF CURRENT MODE RC OSCILLATORS,” *IFAC Proc. Vol.*, vol. 40, no. 18, pp. 547–552, Sep. 2007.
- [26] P. Horowitz and W. Hill, *The art of electronics*. Foundation Books, 1992.
- [27] S. J. Link, J. Sanny, and B. Moebs, “17.3: Speed of Sound - Physics LibreTexts.” [Online]. Available: [https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_University_Physics_\(OpenStax\)/Map%3A_University_Physics_I_-_Mechanics%2C_Sound%2C_Oscillations%2C_and_Waves_\(OpenStax\)/17%3A_Sound/17.03%3A_Speed_of_Sound](https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_University_Physics_(OpenStax)/Map%3A_University_Physics_I_-_Mechanics%2C_Sound%2C_Oscillations%2C_and_Waves_(OpenStax)/17%3A_Sound/17.03%3A_Speed_of_Sound). [Accessed: 18-Dec-2019].
- [28] JetBrains, “PyCharm: the Python IDE for Professional Developers by JetBrains.” [Online]. Available: <https://www.jetbrains.com/pycharm/>. [Accessed: 23-Dec-2019].
- [29] JetBrains, “YouTrack: The Issue Tracking and Project Management Tool for Software Teams.” [Online]. Available: <https://www.jetbrains.com/youtrack/>. [Accessed: 23-Dec-

2019].

- [30] J. T. Krile, A. A. Neuber, J. C. Dickens, and H. G. Krompholz, "DC flashover of a dielectric surface in atmospheric conditions," *IEEE Trans. Plasma Sci.*, vol. 32, no. 5 I, pp. 1828–1834, Oct. 2004.
- [31] Materialize, "MaterializeCSS." [Online]. Available: <https://materializecss.com/>. [Accessed: 22-Dec-2019].
- [32] MaxBotix Inc., "Understanding How Ultrasonic Sensors Work," 2019. [Online]. Available: <https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm>. [Accessed: 18-Dec-2019].
- [33] Mobatek, "MobaXterm free Xserver and tabbed SSH client for Windows." [Online]. Available: <https://mobaxterm.mobatek.net/>. [Accessed: 23-Dec-2019].
- [34] Mulesoft LLC, "What is an API? (Application Programming Interface)." [Online]. Available: <https://www.mulesoft.com/resources/api/what-is-an-api>. [Accessed: 22-Dec-2019].
- [35] C. R. Nave, "Table of Resistivity," 2017. [Online]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbase/Tables/rstiv.html>. [Accessed: 28-Dec-2019].
- [36] NXP, "Reducing Emissions (Including Greenhouse Gases)." [Online]. Available: <https://www.nxp.com/company/our-company/about-nxp/corporate-responsibility/environment/reducing-emissions:GLOBAL-WARMING>. [Accessed: 28-Dec-2019].
- [37] Oficina Catalana del Canvi Climàtic, "Calculadora d'emissions d'una activitat. Canvi climàtic." [Online]. Available: https://canviclimatic.gencat.cat/ca/actua/calculadora_demissions/index.html. [Accessed: 28-Dec-2019].
- [38] V. A. Rakov and M. A. Uman, *Lightning*. Cambridge: Cambridge University Press, 2003.
- [39] Raspberry Pi Foundation, "Power Supply," 2019. [Online]. Available: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>. [Accessed: 08-Dec-2019].
- [40] Raspberry Pi Foundation, "Mechanical Drawings - Raspberry Pi Documentation." [Online]. Available: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/mechanical/README.md>. [Accessed: 16-Dec-2019].
- [41] Raspberry Pi Foundation, "Introducing Raspberry Pi HATs." [Online]. Available: <https://www.raspberrypi.org/blog/introducing-raspberry-pi-hats/>. [Accessed: 16-Dec-2019].
- [42] Raspberry Pi Foundation, "Buster - the new version of Raspbian," 2019. [Online]. Available: <https://www.raspberrypi.org/blog/buster-the-new-version-of-raspbian/>. [Accessed: 16-Dec-2019].

- [43] Red Eléctrica de España, “España cierra 2019 con un 10 % más de potencia instalada de generación renovable.” [Online]. Available: <https://www.ree.es/es/sala-de-prensa/actualidad/notas-de-prensa/2019/12/espana-cierra-2019-con-un-10-mas-de-potencia-instalada-de-generacion-renovable#>. [Accessed: 28-Dec-2019].
- [44] Redis Labs, “Redis.” [Online]. Available: <https://redis.io/>. [Accessed: 22-Dec-2019].
- [45] Revolution Education Ltd, “Ultrasonic Range Finder (SRF005).” [Online]. Available: <http://www.picaxe.com/docs/srf005.pdf>. [Accessed: 16-Dec-2019].
- [46] E. Ritok, *AM radio Lightning detection (tens of kilometers away)*. 2014.
- [47] M. Rouse, “What is daemon?,” 2005. [Online]. Available: <https://whatis.techtarget.com/definition/daemon>. [Accessed: 22-Dec-2019].
- [48] Scrum.org, “What is Scrum?” [Online]. Available: <https://www.scrum.org/resources/what-is-scrum>. [Accessed: 23-Dec-2019].
- [49] Selectra, “Precio del kwh en España: Comparativa de Endesa, Iberdrola y otras.” [Online]. Available: <https://tarifaluzhora.es/info/precio-kwh>. [Accessed: 23-Dec-2019].
- [50] Sociedad Colibri SAS, “Quick PRD40r 1P+N SCHNEIDER ELECTRIC A9L16292.” [Online]. Available: <https://www.manomano.es/p/quick-prd40r-3p-n-schneider-electric-a9l16294-185399>. [Accessed: 28-Dec-2019].
- [51] A. Solem, M. Rocco, and J. Henrik Helmers, “Celery: Distributed Task Queue.” [Online]. Available: <http://www.celeryproject.org/>. [Accessed: 22-Dec-2019].
- [52] TablePlus Inc, “SQLite vs PostgreSQL - Which database to use and why?” [Online]. Available: <https://tableplus.com/blog/2018/08/sqlite-vs-postgresql-which-database-to-use-and-why.html>. [Accessed: 22-Dec-2019].
- [53] Texas Instruments, “L293x Quadruple Half-H Drivers,” 2016. [Online]. Available: <http://www.ti.com/lit/ds/symlink/l293d.pdf>. [Accessed: 16-Dec-2019].
- [54] The Apache Software Foundation, “Apache Web Server.” [Online]. Available: <https://www.apache.org/>. [Accessed: 22-Dec-2019].
- [55] The PostgreSQL Global Development Group, “PostgreSQL: The world’s most advanced open source database.” [Online]. Available: <https://www.postgresql.org/>. [Accessed: 22-Dec-2019].
- [56] M. A. Uman, *The art and science of lightning protection*. Cambridge University Press, 2008.
- [57] R. Visintini, “Rectifiers,” 2006.
- [58] Wattia Innova, “Performance of off-grid PV systems.” p. 1, 2018.
- [59] F. M. White and University of Rhode Island, *Fluid Mechanics*, 4th ed. McGraw-Hill, 1999.